**<http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework>**

**[How do you handle multiple submit buttons in ASP.NET MVC Framework?](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework)**

[](http://engine.adzerk.net/r?e=_bWVkPXpyayZyZWY9Y2FuZGlkYXRlcy13ZmhfbGItd2ZoLXBhamFtYXMifQ&s=YYvSIa7QZ6-pseAhPRvEXXliG5E)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 489 down vote [favorite](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework)  **249** | Is there some easy way to handle multiple submit buttons from the same form? Example:  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <input type="submit" value="Send" />  <input type="submit" value="Cancel" />  <% Html.EndForm(); %>  Any idea how to do this in ASP.NET Framework Beta? All examples I've googled for have single buttons in them.  [html](http://stackoverflow.com/questions/tagged/html) [asp.net](http://stackoverflow.com/questions/tagged/asp.net) [asp.net-mvc](http://stackoverflow.com/questions/tagged/asp.net-mvc) [http-post](http://stackoverflow.com/questions/tagged/http-post) [form-submit](http://stackoverflow.com/questions/tagged/form-submit)   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/q/442704)[improve this question](http://stackoverflow.com/posts/442704/edit) | [edited Nov 18 '15 at 7:33](http://stackoverflow.com/posts/442704/revisions)  [[https://i.stack.imgur.com/AtkOO.jpg?s=32&g=1](http://stackoverflow.com/users/4813600/md-alamin-mahamud)](http://stackoverflow.com/users/4813600/md-alamin-mahamud)  [Md. Alamin Mahamud](http://stackoverflow.com/users/4813600/md-alamin-mahamud)  512315 | asked Jan 14 '09 at 11:58  [[https://www.gravatar.com/avatar/dc374f99f92956f128ef79df36d819e9?s=32&d=identicon&r=PG](http://stackoverflow.com/users/3713/spoike)](http://stackoverflow.com/users/3713/spoike)  [Spoike](http://stackoverflow.com/users/3713/spoike)  46k32113137 | |
|  | add a comment |

**34 Answers**

[active](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework?answertab=active#tab-top) [oldest](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework?answertab=oldest#tab-top) [votes](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework?answertab=votes#tab-top)

1 [2 next](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework?page=2&tab=votes#tab-top)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 366 down vote accepted | Here is a mostly clean attribute-based solution to the multiple submit button issue based heavily on the post and comments from [Maartin Balliauw](http://blog.maartenballiauw.be/post/2009/11/26/Supporting-multiple-submit-buttons-on-an-ASPNET-MVC-view.aspx).  [AttributeUsage(AttributeTargets.Method, AllowMultiple = false, Inherited = true)]  public class MultipleButtonAttribute : ActionNameSelectorAttribute  {  public string Name { get; set; }  public string Argument { get; set; }  public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo)  {  var isValidName = false;  var keyValue = string.Format("{0}:{1}", Name, Argument);  var value = controllerContext.Controller.ValueProvider.GetValue(keyValue);  if (value != null)  {  controllerContext.Controller.ControllerContext.RouteData.Values[Name] = Argument;  isValidName = true;  }  return isValidName;  }  }  razor:  <form action="" method="post">  <input type="submit" value="Save" name="action:Save" />  <input type="submit" value="Cancel" name="action:Cancel" />  </form>  and controller:  [HttpPost]  [MultipleButton(Name = "action", Argument = "Save")]  public ActionResult Save(MessageModel mm) { ... }  [HttpPost]  [MultipleButton(Name = "action", Argument = "Cancel")]  public ActionResult Cancel(MessageModel mm) { ... }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/7111222)[improve this answer](http://stackoverflow.com/posts/7111222/edit) | [edited Oct 25 '13 at 14:50](http://stackoverflow.com/posts/7111222/revisions)  [[https://i.stack.imgur.com/pkw6V.png?s=32&g=1](http://stackoverflow.com/users/70345/ian-kemp)](http://stackoverflow.com/users/70345/ian-kemp)  [Ian Kemp](http://stackoverflow.com/users/70345/ian-kemp)  11k84973 | answered Aug 18 '11 at 17:00  [[https://www.gravatar.com/avatar/11d23f9146c864ef6d1cb0d343acd025?s=32&d=identicon&r=PG](http://stackoverflow.com/users/677420/mkozicki)](http://stackoverflow.com/users/677420/mkozicki)  [mkozicki](http://stackoverflow.com/users/677420/mkozicki)  3,8151814 | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 2 |  | | I found this solution to be the happy marriage of the other techniques used. Works perfectly and doesn't effect localization. – [nameEqualsPNamePrubeGoldberg](http://stackoverflow.com/users/423311/nameequalspnameprubegoldberg" \o "1,223 reputation) [Nov 29 '11 at 22:33](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment10253127_7111222) | | |  |  | | --- | --- | | 15 |  | | Great solution!! Much nore useful than those top scored answers – [Oleksandr Pshenychnyy](http://stackoverflow.com/users/543591/oleksandr-pshenychnyy" \o "4,007 reputation) [Feb 16 '12 at 17:23](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment11752357_7111222) | | |  |  | | --- | --- | | 2 |  | | Yeaaaaaaaa. Works on MVC 4.5. The other dont seem to work. great +1 – [ppumkin](http://stackoverflow.com/users/706363/ppumkin" \o "4,793 reputation) [Jul 18 '13 at 20:35](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment25851340_7111222) | | |  |  | | --- | --- | | 9 |  | | A problem with this approach is that if you attempt to return View(viewmodel) in the case where your model has errors, it will attempt to return a view called Send or depending on what your argument name is. – [Shoe](http://stackoverflow.com/users/1950321/shoe) [Oct 17 '13 at 16:03](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment28808666_7111222) | | |  |  | | --- | --- | | 11 |  | | @Shoe - have just found a similar thing. Ensure you explicitly specify the name of the view to return if you're using this method: return View("Index", viewModel) – [ajbeaven](http://stackoverflow.com/users/161735/ajbeaven" \o "3,290 reputation) [Nov 27 '13 at 21:08](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment30208843_7111222) |   [show **25** more comments](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 382 down vote | Give your submit buttons a name, and then inspect the submitted value in your controller method:  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <input type="submit" name="submitButton" value="Send" />  <input type="submit" name="submitButton" value="Cancel" />  <% Html.EndForm(); %>  posting to  public class MyController : Controller {  public ActionResult MyAction(string submitButton) {  switch(submitButton) {  case "Send":  // delegate sending to another controller action  return(Send());  case "Cancel":  // call another action to perform the cancellation  return(Cancel());  default:  // If they've submitted the form without a submitButton,  // just return the view again.  return(View());  }  }  private ActionResult Cancel() {  // process the cancellation request here.  return(View("Cancelled"));  }  private ActionResult Send() {  // perform the actual send operation here.  return(View("SendConfirmed"));  }  }  EDIT:  To extend this approach to work with localized sites, isolate your messages somewhere else (e.g. compiling a resource file to a strongly-typed resource class)  Then modify the code so it works like:  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <input type="submit" name="submitButton" value="<%= Html.Encode(Resources.Messages.Send)%>" />  <input type="submit" name="submitButton" value="<%=Html.Encode(Resources.Messages.Cancel)%>" />  <% Html.EndForm(); %>  and your controller should look like this:  // Note that the localized resources aren't constants, so  // we can't use a switch statement.  if (submitButton == Resources.Messages.Send) {  // delegate sending to another controller action  return(Send());  } else if (submitButton == Resources.Messages.Cancel) {  // call another action to perform the cancellation  return(Cancel());  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/443047)[improve this answer](http://stackoverflow.com/posts/443047/edit) | [edited Nov 3 '10 at 11:37](http://stackoverflow.com/posts/443047/revisions) | answered Jan 14 '09 at 14:02  [[https://www.gravatar.com/avatar/93ec7cff90c748bc4f4da0c540e68bfc?s=32&d=identicon&r=PG](http://stackoverflow.com/users/5017/dylan-beattie)](http://stackoverflow.com/users/5017/dylan-beattie)  [Dylan Beattie](http://stackoverflow.com/users/5017/dylan-beattie)  32.5k1993151 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 22 |  | | too bad you depend on the text displayed on the button, it's kinda tricky with a multilanguage user interface – [Omu](http://stackoverflow.com/users/112100/omu" \o "24,533 reputation) [Apr 22 '10 at 11:52](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment2711405_443047) | | |  |  | | --- | --- | | 3 |  | | Switch/case only works with constants, so the localized version can't use switch/case. You need to switch to if else or some other dispatch method. – [mlibby](http://stackoverflow.com/users/13468/mlibby" \o "3,667 reputation) [Nov 1 '10 at 20:58](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment4377642_443047) | | |  |  | | --- | --- | | 7 |  | | you should use a <button type="submit" instead of <input type, because the value of a <button type isn't the text ;). Then you can have something like this: <button name="mySubmitButton" type="submit" value="keyValue">YourButtonText</button> – [J4N](http://stackoverflow.com/users/397830/j4n) [May 29 '12 at 12:32](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment14049328_443047) | | |  |  | | --- | --- | | 4 |  | | How would this work with passing the model to the action instead of just the submit value? – [bizzehdee](http://stackoverflow.com/users/2089006/bizzehdee" \o "6,823 reputation) [Aug 7 '13 at 7:56](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment26491693_443047) | | |  |  | | --- | --- | | 2 |  | | Be careful not to name your buttons "action" if you're using jQuery. It causes a conflict within the library that breaks the action URL. – [HotN](http://stackoverflow.com/users/764371/hotn" \o "2,138 reputation) [Oct 14 '14 at 21:20](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment41398223_443047) |   [show **16** more comments](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework) | |
| up vote 107 down vote | | You can check the name in the action as has been mentioned, but you might consider whether or not this is good design. It is a good idea to consider the responsibility of the action and not couple this design too much to UI aspects like button names. So consider using 2 forms and 2 actions:  <% Html.BeginForm("Send", "MyController", FormMethod.Post); %>  <input type="submit" name="button" value="Send" />  <% Html.EndForm(); %>  <% Html.BeginForm("Cancel", "MyController", FormMethod.Post); %>  <input type="submit" name="button" value="Cancel" />  <% Html.EndForm(); %>  Also, in the case of "Cancel", you are usually just not processing the form and are going to a new URL. In this case you do not need to submit the form at all and just need a link:  <%=Html.ActionLink("Cancel", "List", "MyController") %>   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/442762)[improve this answer](http://stackoverflow.com/posts/442762/edit) | answered Jan 14 '09 at 12:26  [[https://www.gravatar.com/avatar/c6ef010ccf99f694969f50e360700ed6?s=32&d=identicon&r=PG](http://stackoverflow.com/users/41783/trevor-de-koekkoek)](http://stackoverflow.com/users/41783/trevor-de-koekkoek)  [Trevor de Koekkoek](http://stackoverflow.com/users/41783/trevor-de-koekkoek)  2,22211413 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 51 |  | | This is ok when you dont need same form data for every submit button. If you need all data in common form than Dylan Beattie is the way to go. Is there any more elegant way to do this? – [zidane](http://stackoverflow.com/users/86054/zidane" \o "448 reputation) [Sep 28 '09 at 9:30](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment1337844_442762) | | |  |  | | --- | --- | | 3 |  | | About visual presentation, how in this case have the "Send" button next to the "Cancel" button ? – [Kris-I](http://stackoverflow.com/users/118584/kris-i) [Mar 9 '11 at 4:49](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment5900147_442762) | | |  |  | | --- | --- | | 1 |  | | Dylan: Well for a cancel button you don't need to submit the data at all and it is bad practice to couple the controller to the UI elements. However if you can make a more or less generic "command" then I think it is ok, but I would not tie it to "submitButton" as that is the name of a UI element. – [Trevor de Koekkoek](http://stackoverflow.com/users/41783/trevor-de-koekkoek) [Mar 16 '11 at 15:08](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment6012630_442762) | | |  |  | | --- | --- | | 1 |  | | @Kris: you can position your buttons with CSS and they can still reside in 2 different form sections. – [Trevor de Koekkoek](http://stackoverflow.com/users/41783/trevor-de-koekkoek) [Mar 16 '11 at 15:10](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment6012663_442762) | | |  |  | | --- | --- | | 7 |  | | seriously? does that not smell to anyone but me?! – [iwayneo](http://stackoverflow.com/users/156888/iwayneo" \o "7,224 reputation) [Nov 24 '11 at 11:48](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment10160149_442762) |   [show **2** more comments](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 83 down vote | | Eilon suggests you can do it like this:  If you have more than one button you can distinguish between them by giving each button a name:  <input type="submit" name="SaveButton" value="Save data" />  <input type="submit" name="CancelButton" value="Cancel and go back to main page" />  In your controller action method you can add parameters named after the HTML input tag names:  public ActionResult DoSomeStuff(string saveButton, string  cancelButton, ... other parameters ...)  { ... }  If any value gets posted to one of those parameters, that means that button was the one that got clicked. The web browser will only post a value for the *one* button that got clicked. All other values will be null.  if (saveButton != null) { /\* do save logic \*/ }  if (cancelButton != null) { /\* do cancel logic \*/ }  I like this method as it does not rely on the value property of the submit buttons which is more likely to change than the assigned names and doesn't require javascript to be enabled  See: <http://forums.asp.net/p/1369617/2865166.aspx#2865166>   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/463204)[improve this answer](http://stackoverflow.com/posts/463204/edit) | [edited Jan 21 '09 at 16:51](http://stackoverflow.com/posts/463204/revisions)  [[https://www.gravatar.com/avatar/dc374f99f92956f128ef79df36d819e9?s=32&d=identicon&r=PG](http://stackoverflow.com/users/3713/spoike)](http://stackoverflow.com/users/3713/spoike)  [Spoike](http://stackoverflow.com/users/3713/spoike)  46k32113137 | answered Jan 20 '09 at 21:30  [[https://www.gravatar.com/avatar/3c155f90f8b32742f6f499937d7f135f?s=32&d=identicon&r=PG](http://stackoverflow.com/users/26147/pabloblamirez)](http://stackoverflow.com/users/26147/pabloblamirez)  [PabloBlamirez](http://stackoverflow.com/users/26147/pabloblamirez)  1,199612 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | If anyone comes across this old question, this is the cleanest answer if you don't wan't to use HTML5 <button> elements. If you don't mind HTML5 then use <button> with value attribute. – [Kugel](http://stackoverflow.com/users/177591/kugel) [Jun 12 '13 at 1:28](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment24662547_463204) | | |  |  | | --- | --- | |  |  | | How would you do it if creating an ajax call on this form. It seems form.serialize() does not pickup up submit button name.. – [mko](http://stackoverflow.com/users/487291/mko" \o "813 reputation) [Sep 15 '14 at 14:15](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment40446442_463204) |   add a comment |
| up vote 35 down vote | Just written a post about that: [Multiple submit buttons with ASP.NET MVC](http://blog.ashmind.com/index.php/2010/03/15/multiple-submit-buttons-with-asp-net-mvc-final-solution/):  Basically, instead of using ActionMethodSelectorAttribute, I am using ActionNameSelectorAttribute, which allows me to pretend the action name is whatever I want it to be. Fortunately, ActionNameSelectorAttribute does not just make me specify action name, instead I can choose whether the current action matches request.  So there is my class (btw I am not too fond of the name):  public class HttpParamActionAttribute : ActionNameSelectorAttribute {  public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo) {  if (actionName.Equals(methodInfo.Name, StringComparison.InvariantCultureIgnoreCase))  return true;  if (!actionName.Equals("Action", StringComparison.InvariantCultureIgnoreCase))  return false;  var request = controllerContext.RequestContext.HttpContext.Request;  return request[methodInfo.Name] != null;  }  }  To use just define a form like this:  <% using (Html.BeginForm("Action", "Post")) { %>  <!— …form fields… -->  <input type="submit" name="saveDraft" value="Save Draft" />  <input type="submit" name="publish" value="Publish" />  <% } %>  and controller with two methods  public class PostController : Controller {  [HttpParamAction]  [AcceptVerbs(HttpVerbs.Post)]  public ActionResult SaveDraft(…) {  //…  }  [HttpParamAction]  [AcceptVerbs(HttpVerbs.Post)]  public ActionResult Publish(…) {  //…  }  }  As you see, the attribute does not require you to specify anything at all. Also, name of the buttons are translated directly to the method names. Additionally (I haven’t tried that) these should work as normal actions as well, so you can post to any of them directly.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/2449717)[improve this answer](http://stackoverflow.com/posts/2449717/edit) | [edited Jul 18 '13 at 3:05](http://stackoverflow.com/posts/2449717/revisions) | answered Mar 15 '10 at 19:03  [[https://www.gravatar.com/avatar/ecc96b01307b339812f5d1b2936ac9bc?s=32&d=identicon&r=PG](http://stackoverflow.com/users/39068/andrey-shchekin)](http://stackoverflow.com/users/39068/andrey-shchekin)  [Andrey Shchekin](http://stackoverflow.com/users/39068/andrey-shchekin)  9,800657110 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 1 |  | | Beautiful! I think this is the most elegant solution. It eliminates the *value* of the submit tag from consideration, which is ideal since it is a pure-UI attribute that should have no bearing on the control-flow. Instead, the unique name attribute of each submit tag translates directly into a discrete action method on your controller. – [Kirk Woll](http://stackoverflow.com/users/189950/kirk-woll) [Oct 26 '11 at 4:29](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment9643140_2449717) | | |  |  | | --- | --- | |  |  | | +1 For me, it's by far the best solution for this problem. Since I implemented it, I notice that a lot of traffic pass throught the HttpParamActionAttribut but compared to all other things that Asp.Net MVC have to do while processing a request, it's totally acceptable. To only hack I have to do is put an empty 'Action' named in my controller to prevent Resharper warning me that the action 'Action' don't exist. Thank you very much! – [Samuel](http://stackoverflow.com/users/289887/samuel) [Jan 18 '13 at 18:39](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment20044836_2449717) | | |  |  | | --- | --- | |  |  | | I reviewed all of the solutions and also agree this is a nice elegant, simple solution. Great bc there are no conditional statements and robust where you can define a new controller action when you have a new button. Called my class MultiButtonActionHandler FYI ;-) – [ejhost](http://stackoverflow.com/users/2395030/ejhost" \o "114 reputation) [Apr 21 '15 at 19:30](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment47692889_2449717) |   add a comment | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 32 down vote | I would suggest interested parties have a look at a solution i think is very elegant here >> <http://blog.maartenballiauw.be/post/2009/11/26/Supporting-multiple-submit-buttons-on-an-ASPNET-MVC-view.aspx>  In case the link dissapears its using the MultiButton attribute applied to a controller action to indicate which button click that action should relate to.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/2692230)[improve this answer](http://stackoverflow.com/posts/2692230/edit) | [edited Nov 2 '10 at 15:29](http://stackoverflow.com/posts/2692230/revisions)  [[https://www.gravatar.com/avatar/5dc355175b5c20105f04a868fa42562c?s=32&d=identicon&r=PG](http://stackoverflow.com/users/1127460/simon-keep)](http://stackoverflow.com/users/1127460/simon-keep)  [Simon Keep](http://stackoverflow.com/users/1127460/simon-keep)  5,50364772 | answered Apr 22 '10 at 15:36  [[https://www.gravatar.com/avatar/042ccf8a566f356a2f2509d863db596c?s=32&d=identicon&r=PG](http://stackoverflow.com/users/60172/izmoto)](http://stackoverflow.com/users/60172/izmoto)  [Izmoto](http://stackoverflow.com/users/60172/izmoto)  1,39411119 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 1 |  | | +1 Lovely solution .. quite crisp and clean! – [Rashmi Pandit](http://stackoverflow.com/users/39289/rashmi-pandit) [Aug 23 '10 at 5:53](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment3712579_2692230) | | |  |  | | --- | --- | |  |  | | This is the solution we are using now and its very neat. Is it MVC 2 only though? – [Simon Keep](http://stackoverflow.com/users/1127460/simon-keep) [Nov 2 '10 at 15:30](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment4385339_2692230) | | |  |  | | --- | --- | |  |  | | This is beautiful! I'd not seen this before! While I agree that you may want to redesign any solution that is using multiple submits to only use one button, I'm in a spot where I'm hamstrung and must do this. This answer should have won! – [Rikon](http://stackoverflow.com/users/710273/rikon" \o "1,621 reputation) [Sep 30 '11 at 14:19](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment9237461_2692230) | | |  |  | | --- | --- | |  |  | | This is a great solution. Very clean – [Arnej65](http://stackoverflow.com/users/193122/arnej65) [Oct 24 '11 at 13:42](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment9610955_2692230) | | |  |  | | --- | --- | | 1 |  | | Agree this should win. – [daddywoodland](http://stackoverflow.com/users/153782/daddywoodland" \o "1,121 reputation) [Nov 15 '11 at 12:20](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment9983316_2692230) |   [show **2** more comments](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework) | |
| up vote 15 down vote | | You should be able to name the buttons and give them a value; then map this name as an argument to the action. Alternatively, use 2 separate action-links or 2 forms.   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/442721)[improve this answer](http://stackoverflow.com/posts/442721/edit) | answered Jan 14 '09 at 12:07  [[https://i.stack.imgur.com/NJcqr.png?s=32&g=1](http://stackoverflow.com/users/23354/marc-gravell)](http://stackoverflow.com/users/23354/marc-gravell)  [Marc Gravell](http://stackoverflow.com/users/23354/marc-gravell)♦  612k14017132186 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | By far the cleanest, and easiest, solution I've seen. – [Jason Evans](http://stackoverflow.com/users/127440/jason-evans) [Feb 10 '15 at 11:11](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment45191181_442721) |   add a comment |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 9 down vote | You could write:  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <input type="submit" name="button" value="Send" />  <input type="submit" name="button" value="Cancel" />  <% Html.EndForm(); %>  And then in the page check if the name == "Send" or name == "Cancel"...   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/442731)[improve this answer](http://stackoverflow.com/posts/442731/edit) | [edited Aug 13 '10 at 12:39](http://stackoverflow.com/posts/442731/revisions)  [[https://www.gravatar.com/avatar/11a897f9c594e8256f1d550997863adf?s=32&d=identicon&r=PG](http://stackoverflow.com/users/90011/xiii)](http://stackoverflow.com/users/90011/xiii)  [XIII](http://stackoverflow.com/users/90011/xiii)  11.5k52248 | answered Jan 14 '09 at 12:11  [[https://www.gravatar.com/avatar/22835fabf4fc28d990ebc46e78b767a5?s=32&d=identicon&r=PG](http://stackoverflow.com/users/47213/ironicnet)](http://stackoverflow.com/users/47213/ironicnet)  [Ironicnet](http://stackoverflow.com/users/47213/ironicnet)  452312 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 1 |  | | Although this is working, but I think it's wrong practice to have two elements with the same name. – [Péter](http://stackoverflow.com/users/465608/p%c3%a9ter" \o "1,219 reputation) [Jul 5 '13 at 11:25](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment25416497_442731) | | |  |  | | --- | --- | | 1 |  | | It's not necessary wrong. It depends of how you are using the inputs. You can have multiple elements with the same name, and expecting to receive multiple data (this is how the radiobuttons and checkboxes works). But yeah, if you are using this method is because you are doing it "wrong"... That's why i put "You could" but not "You should" :P – [Ironicnet](http://stackoverflow.com/users/47213/ironicnet" \o "452 reputation) [Aug 30 '13 at 18:29](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment27268152_442731) |   add a comment | |
| up vote 8 down vote | | Here is what works best for me:  <input type="submit" value="Delete" name="onDelete" />  <input type="submit" value="Save" name="onSave" />  public ActionResult Practice(MyModel model, string onSave, string onDelete)  {  if (onDelete != null)  {  // Delete the object  ...  return EmptyResult();  }  // Save the object  ...  return EmptyResult();  }   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/8598195)[improve this answer](http://stackoverflow.com/posts/8598195/edit) | answered Dec 22 '11 at 0:46  [[https://www.gravatar.com/avatar/89aaca5af4e9325d5e11386adae0f022?s=32&d=identicon&r=PG](http://stackoverflow.com/users/715544/sergey)](http://stackoverflow.com/users/715544/sergey)  [Sergey](http://stackoverflow.com/users/715544/sergey)  10112 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | I get null value for onDelete and onSave in the controller method. Do you know why? – [Diganta Kumar](http://stackoverflow.com/users/798727/diganta-kumar" \o "2,389 reputation) [Jan 24 '13 at 2:30](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment20197993_8598195) | | |  |  | | --- | --- | |  |  | | Either one won't be null if you click the according button. Which button do you click getting the null? – [Sergey](http://stackoverflow.com/users/715544/sergey) [Jul 8 '13 at 17:50](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment25497501_8598195) |   add a comment |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 8 down vote | Something I don't like about ActionSelectName is that IsValidName is called for every action method in the controller; I don't know why it works this way. I like a solution where every button has a different name based on what it does, but I don't like the fact that you have to have as many parameters in the action method as buttons in the form. I have created an enum for all button types:  public enum ButtonType  {  Submit,  Cancel,  Delete  }  Instead of ActionSelectName, I use an ActionFilter:  public class MultipleButtonsEnumAttribute : ActionFilterAttribute  {  public Type EnumType { get; set; }  public MultipleButtonsEnumAttribute(Type enumType)  {  EnumType = enumType;  }  public override void OnActionExecuting(ActionExecutingContext filterContext)  {  foreach (var key in filterContext.HttpContext.Request.Form.AllKeys)  {  if (Enum.IsDefined(EnumType, key))  {  var pDesc = filterContext.ActionDescriptor.GetParameters()  .FirstOrDefault(x => x.ParameterType == EnumType);  filterContext.ActionParameters[pDesc.ParameterName] = Enum.Parse(EnumType, key);  break;  }  }  }  }  The filter will find the button name in the form data and if the button name matches any of the button types defined in the enum, it will find the ButtonType parameter among the action parameters:  [MultipleButtonsEnumAttribute(typeof(ButtonType))]  public ActionResult Manage(ButtonType buttonPressed, ManageViewModel model)  {  if (button == ButtonType.Cancel)  {  return RedirectToAction("Index", "Home");  }  //and so on  return View(model)  }  and then in views, I can use:  <input type="submit" value="Button Cancel" name="@ButtonType.Cancel" />  <input type="submit" value="Button Submit" name="@ButtonType.Submit" />   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/20594182)[improve this answer](http://stackoverflow.com/posts/20594182/edit) | [edited Dec 5 '14 at 17:56](http://stackoverflow.com/posts/20594182/revisions)  [[https://www.gravatar.com/avatar/9cb475207ee062662aae4c80ba2fb2c7?s=32&d=identicon&r=PG](http://stackoverflow.com/users/15778/gordon-mackie-joanmiro)](http://stackoverflow.com/users/15778/gordon-mackie-joanmiro)  [Gordon Mackie JoanMiro](http://stackoverflow.com/users/15778/gordon-mackie-joanmiro)  2,66632333 | answered Dec 15 '13 at 11:53  [[https://i.stack.imgur.com/s63pY.jpg?s=32&g=1](http://stackoverflow.com/users/300659/sanjuro)](http://stackoverflow.com/users/300659/sanjuro)  [sanjuro](http://stackoverflow.com/users/300659/sanjuro)  63611022 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | I liked your idea, thanks! – [Stefano Magistri](http://stackoverflow.com/users/1863856/stefano-magistri) [Dec 15 '15 at 16:20](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment56329950_20594182) |   add a comment | |
| up vote 5 down vote | | David Findley writes about 3 different options you have for doing this, on his ASP.Net weblog.  Read the article [multiple buttons in the same form](http://weblogs.asp.net/dfindley/archive/2009/05/31/asp-net-mvc-multiple-buttons-in-the-same-form.aspx) to see his solutions, and the advantages and disadvantages of each. IMHO he provides a very elegant solution which makes use of attributes that you decorate your action with.   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/1048229)[improve this answer](http://stackoverflow.com/posts/1048229/edit) | answered Jun 26 '09 at 9:35  [[https://www.gravatar.com/avatar/bfcdf385b88edc7332c38557c3b489dc?s=32&d=identicon&r=PG](http://stackoverflow.com/users/127488/saajid-ismail)](http://stackoverflow.com/users/127488/saajid-ismail)  [Saajid Ismail](http://stackoverflow.com/users/127488/saajid-ismail)  2,78083251 | |
|  | | add a comment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 5 down vote | This is the technique I'd use and I don't see it here yet. The link (posted by Saajid Ismail ) that inspires this solution is <http://weblogs.asp.net/dfindley/archive/2009/05/31/asp-net-mvc-multiple-buttons-in-the-same-form.aspx>). It adapts Dylan Beattie's answer to do localization without any problems.  In the View:  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <button name="button" value="send"><%: Resources.Messages.Send %></button>  <button name="button" value="cancel"><%: Resources.Messages.Cancel %></button>  <% Html.EndForm(); %>  In the Controller:  public class MyController : Controller  {  public ActionResult MyAction(string button)  {  switch(button)  {  case "send":  this.DoSend();  break;  case "cancel":  this.DoCancel();  break;  }  }  }   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/4073151)[improve this answer](http://stackoverflow.com/posts/4073151/edit) | answered Nov 1 '10 at 21:20  [[https://www.gravatar.com/avatar/26a81e5badb9e002ab9ed3542036e584?s=32&d=identicon&r=PG](http://stackoverflow.com/users/13468/mlibby)](http://stackoverflow.com/users/13468/mlibby)  [mlibby](http://stackoverflow.com/users/13468/mlibby)  3,66711629 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Looks like the solution Ironicnet provided. – [XIII](http://stackoverflow.com/users/90011/xiii) [Nov 2 '10 at 15:02](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment4385006_4073151) | | |  |  | | --- | --- | |  |  | | Certainly similar, but this shows both localization and the controller code, which is something I didn't see done this way in this thread. I found this thread while looking for how to do this and wanted to document what I came up with for anyone else who might be in the same boat. – [mlibby](http://stackoverflow.com/users/13468/mlibby" \o "3,667 reputation) [Nov 3 '10 at 11:56](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment4394856_4073151) | | |  |  | | --- | --- | | 1 |  | | In fact, it's not the same as Ironicnet's beyond that. He uses <input> elements. I use <button>, which is required to do the localization without having variable value attributes. – [mlibby](http://stackoverflow.com/users/13468/mlibby" \o "3,667 reputation) [Nov 3 '10 at 11:59](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment4394879_4073151) |   add a comment | |
| up vote 5 down vote | | This script allows to specify a data-form-action attribute which will work as the HTML5 formaction attribute in all browsers (in an unobtrusive way):  $(document).on('click', '[type="submit"][data-form-action]', function(event) {  var $this = $(this),  var formAction = $this.attr('data-form-action'),  $form = $($this.closest('form'));  $form.attr('action', formAction);  });  The form containing the button will be posted to the URL specified in the data-form-action attribute:  <button type="submit" data-form-action="different/url">Submit</button>  This requires jQuery 1.7. For previous versions you should use live() instead of on().   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/9451118)[improve this answer](http://stackoverflow.com/posts/9451118/edit) | [edited May 16 '13 at 12:16](http://stackoverflow.com/posts/9451118/revisions)  [[https://i.stack.imgur.com/SluAb.jpg?s=32&g=1](http://stackoverflow.com/users/1216612/jotabe)](http://stackoverflow.com/users/1216612/jotabe)  [JotaBe](http://stackoverflow.com/users/1216612/jotabe)  21.9k23867 | answered Feb 26 '12 at 6:39  [[https://www.gravatar.com/avatar/e867260f1ac344d64d25bc1e3fb1910e?s=32&d=identicon&r=PG](http://stackoverflow.com/users/463042/jay-douglass)](http://stackoverflow.com/users/463042/jay-douglass)  [Jay Douglass](http://stackoverflow.com/users/463042/jay-douglass)  3,99811616 | |
|  | | add a comment |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 5 down vote | There are three ways by which you can solve the above issue   1. HTML way 2. Jquery way 3. “ActionNameSelectorAttribute” way   Below is a video which summarizes all the three approaches in a demonstrative way.  <https://www.facebook.com/shivprasad.koirala/videos/vb.100002224977742/809335512483940>  **HTML way :-**  In the HTML way we need to create two forms and place the “Submit” button inside each of the forms. And every form’s action will point to different / respective actions. You can see the below code the first form is posting to “Action1” and the second form will post to “Action2” depending on which “Submit” button is clicked.  <form action="Action1" method=post>  <input type=”submit” name=”Submit1”/>  </form>  <form action="Action2" method=post>  <input type=”submit” name=”Submit2”>  </form>  **Ajax way :-**  In case you are a Ajax lover this second option would excite you more. In the Ajax way we can create two different functions “Fun1” and “Fun1” , see the below code. These functions will make Ajax calls by using JQUERY or any other framework. Each of these functions are binded with the “Submit” button’s “OnClick” events. Each of these function make call to respective action names.  <Script language="javascript">  function Fun1()  {  $.post(“/Action1”,null,CallBack1);  }  function Fun2()  {  $.post(“/Action2”,null,CallBack2);  }  </Script>  <form action="/Action1" method=post>  <input type=submit name=sub1 onclick=”Fun2()”/>  </form>  <form action="/Action2" method=post>  <input type=submit name=sub2 onclick=”Fun1()”/>  </form>  **Using “ActionNameSelectorAttribute”:-**  This is a great and a clean option. The “ActionNameSelectorAttribute” is a simple attribute class where we can write decision making logic which will decide which action can be executed.  So the first thing is in HTML we need to put proper name’s to the submit buttons for identifying them on the server.  You can see we have put “Save” and “Delete” to the button names. Also you can notice in the action we have just put controller name “Customer” and not a particular action name. We expect the action name will be decide by “ActionNameSelectorAttribute”.  <form action=”Customer” method=post>  <input type=submit value="Save" name="Save" /> <br />  <input type=submit value="Delete" name="Delete"/>  </form>  So when the submit button is clicked , it first hits the “ActionNameSelector” attribute and then depending on which submit is fired it invokes the appropriate action.  enter image description here  So the first step is to create a class which inherits from “ActionNameSelectorAttribute” class. In this class we have created a simple property “Name”.  We also need to override the “IsValidName” function which returns true or flase. This function is where we write the logic whether an action has to be executed or not. So if this function returns true then the action is executed or else it is not.  public class SubmitButtonSelector : ActionNameSelectorAttribute  {  public string Name { get; set; }  public override bool IsValidName(ControllerContext controllerContext, string actionName, System.Reflection.MethodInfo methodInfo)  {  // Try to find out if the name exists in the data sent from form  var value = controllerContext.Controller.ValueProvider.GetValue(Name);  if (value != null)  {  return true;  }  return false;  }  }  The main heart of the above function is in the below code. The “ValueProvider” collection has all the data that has been posted from the form. So it first looks up the “Name” value and if its found in the HTTP request it returns true or else it returns false.  var value = controllerContext.Controller.ValueProvider.GetValue(Name);  if (value != null)  {  return true;  }  return false;  This attribute class can then decorated on the respective action and the respective “Name” value can be provided. So if the submit is hitting this action and if the name matches of the HTML submit button name it then executes the action further or else it does not.  public class CustomerController : Controller  {  [SubmitButtonSelector(Name="Save")]  public ActionResult Save()  {  return Content("Save Called");  }  [SubmitButtonSelector(Name = "Delete")]  public ActionResult Delete()  {  return Content("Delete Called");  }  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/30352851)[improve this answer](http://stackoverflow.com/posts/30352851/edit) | [edited May 21 '15 at 3:29](http://stackoverflow.com/posts/30352851/revisions) | answered May 20 '15 at 14:37  [[https://www.gravatar.com/avatar/f3702a13fcfab9c2a6750be11b6e7087?s=32&d=identicon&r=PG](http://stackoverflow.com/users/993672/shivprasad-koirala)](http://stackoverflow.com/users/993672/shivprasad-koirala)  [Shivprasad Koirala](http://stackoverflow.com/users/993672/shivprasad-koirala)  6,46413729 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Detailed & informative answer. Very useful. – [Funk247](http://stackoverflow.com/users/1978095/funk247) [Aug 18 '15 at 14:00](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment52045541_30352851) |   add a comment | |
| up vote 4 down vote | | Here's an extension method I wrote to handle multiple image and/or text buttons.  Here's the HTML for an image button:  <input id="btnJoin" name="Join" src="/content/images/buttons/btnJoin.png"  type="image">  or for a text submit button :  <input type="submit" class="ui-button green" name="Submit\_Join" value="Add to cart" />  <input type="submit" class="ui-button red" name="Submit\_Skip" value="Not today" />  Here is the extension method you call from the controller with form.GetSubmitButtonName(). For image buttons it looks for a form parameter with .x (which indicates an image button was clicked) and extracts the name. For regular input buttons it looks for a name beginning with Submit\_ and extracts the command from afterwards. Because I'm abstracting away the logic of determining the 'command' you can switch between image + text buttons on the client without changing the server side code.  public static class FormCollectionExtensions  {  public static string GetSubmitButtonName(this FormCollection formCollection)  {  return GetSubmitButtonName(formCollection, true);  }  public static string GetSubmitButtonName(this FormCollection formCollection, bool throwOnError)  {  var imageButton = formCollection.Keys.OfType<string>().Where(x => x.EndsWith(".x")).SingleOrDefault();  var textButton = formCollection.Keys.OfType<string>().Where(x => x.StartsWith("Submit\_")).SingleOrDefault();  if (textButton != null)  {  return textButton.Substring("Submit\_".Length);  }  // we got something like AddToCart.x  if (imageButton != null)  {  return imageButton.Substring(0, imageButton.Length - 2);  }  if (throwOnError)  {  throw new ApplicationException("No button found");  }  else  {  return null;  }  }  }  **Note:** For text buttons you have to prefix the name with Submit\_. I prefer this way becuase it means you can change the text (display) value without having to change the code. Unlike SELECT elements, an INPUT button has only a 'value' and no separate 'text' attribute. My buttons say different things under different contexts - but map to the same 'command'. I much prefer extracting the name this way than having to code for == "Add to cart".   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/5048093)[improve this answer](http://stackoverflow.com/posts/5048093/edit) | [edited Feb 19 '11 at 1:52](http://stackoverflow.com/posts/5048093/revisions) | answered Feb 19 '11 at 0:49  [[https://www.gravatar.com/avatar/7fc2db839382b30434539fbc967c498e?s=32&d=identicon&r=PG](http://stackoverflow.com/users/16940/simon-weaver)](http://stackoverflow.com/users/16940/simon-weaver)  [Simon\_Weaver](http://stackoverflow.com/users/16940/simon-weaver)  48.4k50331434 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | I like checking the name as an alternative because you can't always check for the value for eg. you have a list of items and each has a "Delete" button. – [Max](http://stackoverflow.com/users/165394/max) [Aug 26 '11 at 11:09](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment8653557_5048093) |   add a comment |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 4 down vote | I don't have enough rep to comment in the correct place, but I spent all day on this so want to share.  While trying to implement the "MultipleButtonAttribute" solution ValueProvider.GetValue(keyValue) would incorrectly come back null.  It turned out I was referencing System.Web.MVC version 3.0 when it should have been 4.0 (other assemblies are 4.0). I don't know why my project didn't upgrade correctly and I had no other obvious problems.  So if your ActionNameSelectorAttribute is not working... check that.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/13881969)[improve this answer](http://stackoverflow.com/posts/13881969/edit) | [edited Dec 14 '12 at 16:19](http://stackoverflow.com/posts/13881969/revisions)  [[https://www.gravatar.com/avatar/a2b33da480d9c6a4a9d89ddf996ed77b?s=32&d=identicon&r=PG](http://stackoverflow.com/users/592301/lhan)](http://stackoverflow.com/users/592301/lhan)  [lhan](http://stackoverflow.com/users/592301/lhan)  2,83353380 | answered Dec 14 '12 at 15:50  [[https://www.gravatar.com/avatar/d9a4804580a3e204366ac4ce75791407?s=32&d=identicon&r=PG](http://stackoverflow.com/users/1416270/heatherd)](http://stackoverflow.com/users/1416270/heatherd)  [HeatherD](http://stackoverflow.com/users/1416270/heatherd)  866 | | |
|  | add a comment | |
| up vote 3 down vote | | If you do not have restrictions on the use of HTML 5, you can use the <button> tag with formaction Attribute:  <form action="demo\_form.asp" method="get">  First name: <input type="text" name="fname" /><br />  Last name: <input type="text" name="lname" /><br />  <button type="submit">Submit</button><br />  <button type="submit" formaction="demo\_admin.asp">Submit as admin</button>  </form>  Reference: <http://www.w3schools.com/html5/att_button_formaction.asp>   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/8049172)[improve this answer](http://stackoverflow.com/posts/8049172/edit) | [edited Nov 8 '11 at 10:51](http://stackoverflow.com/posts/8049172/revisions) | answered Nov 8 '11 at 10:43  [[https://www.gravatar.com/avatar/2bd48ba52943d88eb065c698c1ba807e?s=32&d=identicon&r=PG](http://stackoverflow.com/users/492460/acaz-souza)](http://stackoverflow.com/users/492460/acaz-souza)  [Acaz Souza](http://stackoverflow.com/users/492460/acaz-souza)  3,14153468 | |
|  | | add a comment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 3 down vote | | I've tried to make a synthesis of all solutions and created a [ButtenHandler] attribute that makes it easy to handle multiple buttons on a form.  I've described it on CodeProject [Multiple parameterized (localizable) form buttons in ASP.NET MVC](http://www.codeproject.com/Articles/464178/Multiple-parameterized-localizable-form-buttons-in).  To handle the simple case of this button:  <button type="submit" name="AddDepartment">Add Department</button>  You'll have something like the following action method:  [ButtonHandler()]  public ActionResult AddDepartment(Company model)  {  model.Departments.Add(new Department());  return View(model);  }  Notice how the name of the button matches the name of the action method. The article also describes how to have buttons with values and buttons with indexes.   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/12606378)[improve this answer](http://stackoverflow.com/posts/12606378/edit) | answered Sep 26 '12 at 16:34  [[https://www.gravatar.com/avatar/00296aedc9cb2148c262cf130d1e8dfc?s=32&d=identicon&r=PG](http://stackoverflow.com/users/323122/codetuner)](http://stackoverflow.com/users/323122/codetuner)  [codetuner](http://stackoverflow.com/users/323122/codetuner)  1125 | |
|  | | add a comment |
| up vote 3 down vote | Modified version of HttpParamActionAttribute method but with a bug fix for not causing an error on expired/invalid session postbacks. To see if this is a problem with your current site, open the your form in a window and just before you go to click Save or Publish, open a duplicate window, and logout. Now go back to your first window and try to submit your form using either button. For me I got an error so this change solves that problem for me. I omit a bunch of stuff for the sake of brevity but you should get the idea. The key parts are the inclusion of ActionName on the attribute and making sure the name passed in is the name of the View that shows the form  **Attribute Class**  [AttributeUsage(AttributeTargets.Method, AllowMultiple = false, Inherited = true)]  public class HttpParamActionAttribute : ActionNameSelectorAttribute  {  private readonly string actionName;  public HttpParamActionAttribute(string actionName)  {  this.actionName = actionName;  }  public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo)  {  if (actionName.Equals(methodInfo.Name, StringComparison.InvariantCultureIgnoreCase))  return true;  if (!actionName.Equals(this.actionName, StringComparison.InvariantCultureIgnoreCase))  return false;  var request = controllerContext.RequestContext.HttpContext.Request;  return request[methodInfo.Name] != null;  }  }  **Controller**  [Authorize(Roles="CanAddContent")]  public ActionResult CreateContent(Guid contentOwnerId)  {  var viewModel = new ContentViewModel  {  ContentOwnerId = contentOwnerId  //populate rest of view model  }  return View("CreateContent", viewModel);  }  [Authorize(Roles="CanAddContent"), HttpPost, HttpParamAction("CreateContent"), ValidateAntiForgeryToken]  public ActionResult SaveDraft(ContentFormModel model)  {  //Save as draft  return RedirectToAction("CreateContent");  }  [Authorize(Roles="CanAddContent"), HttpPost, HttpParamAction("CreateContent"), ValidateAntiForgeryToken]  public ActionResult Publish(ContentFormModel model)  {  //publish content  return RedirectToAction("CreateContent");  }  **View**  @using (Ajax.BeginForm("CreateContent", "MyController", new { contentOwnerId = Model.ContentOwnerId }))  {  @Html.AntiForgeryToken()  @Html.HiddenFor(x => x.ContentOwnerId)  <!-- Rest of your form controls -->  <input name="SaveDraft" type="submit" value="SaveDraft" />  <input name="Publish" type="submit" value="Publish" />  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/19434265)[improve this answer](http://stackoverflow.com/posts/19434265/edit) | [edited Oct 17 '13 at 18:25](http://stackoverflow.com/posts/19434265/revisions) | answered Oct 17 '13 at 18:19  [[https://www.gravatar.com/avatar/383b31276adfcd2a4410680852e58c79?s=32&d=identicon&r=PG](http://stackoverflow.com/users/466224/nick-albrecht)](http://stackoverflow.com/users/466224/nick-albrecht)  [Nick Albrecht](http://stackoverflow.com/users/466224/nick-albrecht)  8,63853669 | | |
|  | add a comment | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 3 down vote | this is the best way that i have found:  <http://iwayneo.blogspot.co.uk/2013/10/aspnet-mvc-action-selector-with-list.html>  Here is the code:  /// <summary>  /// ActionMethodSelector to enable submit buttons to execute specific action methods.  /// </summary>  public class AcceptParameterAttribute : ActionMethodSelectorAttribute  {  /// <summary>  /// Gets or sets the value to use to inject the index into  /// </summary>  public string TargetArgument { get; set; }  /// <summary>  /// Gets or sets the value to use in submit button to identify which method to select. This must be unique in each controller.  /// </summary>  public string Action { get; set; }  /// <summary>  /// Gets or sets the regular expression to match the action.  /// </summary>  public string ActionRegex { get; set; }  /// <summary>  /// Determines whether the action method selection is valid for the specified controller context.  /// </summary>  /// <param name="controllerContext">The controller context.</param>  /// <param name="methodInfo">Information about the action method.</param>  /// <returns>true if the action method selection is valid for the specified controller context; otherwise, false.</returns>  public override bool IsValidForRequest(ControllerContext controllerContext, MethodInfo methodInfo)  {  if (controllerContext == null)  {  throw new ArgumentNullException("controllerContext");  }  Func<NameValueCollection> formGetter;  Func<NameValueCollection> queryStringGetter;  ValidationUtility.GetUnvalidatedCollections(HttpContext.Current, out formGetter, out queryStringGetter);  var form = formGetter();  var queryString = queryStringGetter();  var req = form.AllKeys.Any() ? form : queryString;  if (!string.IsNullOrEmpty(this.ActionRegex))  {  foreach (var key in req.AllKeys.Where(k => k.StartsWith(Action, true, System.Threading.Thread.CurrentThread.CurrentCulture)))  {  if (key.Contains(":"))  {  if (key.Split(':').Count() == this.ActionRegex.Split(':').Count())  {  bool match = false;  for (int i = 0; i < key.Split(':').Count(); i++)  {  if (Regex.IsMatch(key.Split(':')[0], this.ActionRegex.Split(':')[0]))  {  match = true;  }  else  {  match = false;  break;  }  }  if (match)  {  return !string.IsNullOrEmpty(req[key]);  }  }  }  else  {  if (Regex.IsMatch(key, this.Action + this.ActionRegex))  {  return !string.IsNullOrEmpty(req[key]);  }  }  }  return false;  }  else  {  return req.AllKeys.Contains(this.Action);  }  }  }  Enjoy a code-smell-less multi submit button future.  thank you   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/8256551)[improve this answer](http://stackoverflow.com/posts/8256551/edit) | [edited Oct 28 '13 at 10:57](http://stackoverflow.com/posts/8256551/revisions) | answered Nov 24 '11 at 11:49  [[https://www.gravatar.com/avatar/8793479b6bca51f2a6a11e140608dcb8?s=32&d=identicon&r=PG](http://stackoverflow.com/users/156888/iwayneo)](http://stackoverflow.com/users/156888/iwayneo)  [iwayneo](http://stackoverflow.com/users/156888/iwayneo)  7,224854119 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Link currently broken, this is the closest archived version I could find: [web.archive.org/web/20110706230408/http://blogs.sonatribe.com/…](http://web.archive.org/web/20110706230408/http:/blogs.sonatribe.com/wayne/2011/06/15/171/) – [Ian Kemp](http://stackoverflow.com/users/70345/ian-kemp) [Oct 25 '13 at 15:09](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment29080932_8256551) | | |  |  | | --- | --- | |  |  | | Hi Ian - thanks for digging that out - I've reposted it here: [iwayneo.blogspot.co.uk/2013/10/…](http://iwayneo.blogspot.co.uk/2013/10/aspnet-mvc-action-selector-with-list.html) – [iwayneo](http://stackoverflow.com/users/156888/iwayneo" \o "7,224 reputation) [Oct 28 '13 at 10:57](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment29147932_8256551) |   add a comment | |
| up vote 3 down vote | | I've came across this 'problem' as well but found a rather logical solution by adding the name attribute. I couldn't recall having this problem in other languages.  <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.2>   * ... * If a form contains more than one submit button, only the activated submit button is successful. * ...   Meaning the following code value attributes can be changed, localized, internationalized without the *need* for extra code checking strongly-typed resources files or constants.  <% Html.BeginForm("MyAction", "MyController", FormMethod.Post); %>  <input type="submit" name="send" value="Send" />  <input type="submit" name="cancel" value="Cancel" />  <input type="submit" name="draft" value="Save as draft" />  <% Html.EndForm(); %>`  On the receiving end you would only need to check if any of your known submit types isn't null  public ActionResult YourAction(YourModel model) {  if(Request["send"] != null) {  // we got a send  }else if(Request["cancel"]) {  // we got a cancel, but would you really want to post data for this?  }else if(Request["draft"]) {  // we got a draft  }  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/23387287)[improve this answer](http://stackoverflow.com/posts/23387287/edit) | [edited Jan 9 '15 at 9:42](http://stackoverflow.com/posts/23387287/revisions) | answered Apr 30 '14 at 11:48  [[https://www.gravatar.com/avatar/e010ad5968340806203c39f59349810f?s=32&d=identicon&r=PG](http://stackoverflow.com/users/545019/tom-hofman)](http://stackoverflow.com/users/545019/tom-hofman)  [Tom Hofman](http://stackoverflow.com/users/545019/tom-hofman)  15411 | |
|  | | add a comment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 2 down vote | //model  public class input\_element  {  public string Btn { get; set; }  }  //views--submit btn can be input type also...  @using (Html.BeginForm())  {  <button type="submit" name="btn" value="verify">  Verify data</button>  <button type="submit" name="btn" value="save">  Save data</button>  <button type="submit" name="btn" value="redirect">  Redirect</button>  }  //controller  public ActionResult About()  {  ViewBag.Message = "Your app description page.";  return View();  }  [HttpPost]  public ActionResult About(input\_element model)  {  if (model.Btn == "verify")  {  // the Verify button was clicked  }  else if (model.Btn == "save")  {  // the Save button was clicked  }  else if (model.Btn == "redirect")  {  // the Redirect button was clicked  }  return View();  }   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/15082059)[improve this answer](http://stackoverflow.com/posts/15082059/edit) | answered Feb 26 '13 at 5:33  [[https://www.gravatar.com/avatar/a275b489e58e9bf8fcbc18ddf89008a4?s=32&d=identicon&r=PG](http://stackoverflow.com/users/1300758/shauraj-singh)](http://stackoverflow.com/users/1300758/shauraj-singh)  [SHAURAJ SINGH](http://stackoverflow.com/users/1300758/shauraj-singh)  227317 | | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | You may not have realized it, but this same answer was posted quite a few times to this question already. – [Andrew Barber](http://stackoverflow.com/users/237838/andrew-barber) [Feb 26 '13 at 5:39](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment21212533_15082059) | | |  |  | | --- | --- | | 1 |  | | Also, you seem to post answers that only contain code, with no explanation. Would you consider adding some narrative to explain why the code works, and what makes it an answer to the question? This would be very helpful to the person asking the question, and anyone else who comes along. – [Andrew Barber](http://stackoverflow.com/users/237838/andrew-barber) [Feb 26 '13 at 5:41](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment21212577_15082059) | | |  |  | | --- | --- | |  |  | | i have tested this and its working fine.. – [SHAURAJ SINGH](http://stackoverflow.com/users/1300758/shauraj-singh) [Feb 26 '13 at 5:47](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment21212708_15082059) | | |  |  | | --- | --- | | 1 |  | | Sure, it works fine. But that answer has *already been given by others*, a long time ago. And they included explanations about *why* it works, too. – [Andrew Barber](http://stackoverflow.com/users/237838/andrew-barber) [Feb 26 '13 at 5:49](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment21212753_15082059) |   add a comment | |
| up vote 2 down vote | | If your browser supports the attribute formaction for input buttons (IE 10+, not sure about other browsers) then the following should work:  @using (Html.BeginForm()){  //put form inputs here  <input id="sendBtn" value="Send" type="submit" formaction="@Url.Action("Name Of Send Action")" />  <input id="cancelBtn" value="Cancel" type="submit" formaction="@Url.Action("Name of Cancel Action") />  }   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/22073362)[improve this answer](http://stackoverflow.com/posts/22073362/edit) | answered Feb 27 '14 at 15:44  [[https://www.gravatar.com/avatar/15f2f37e72750d36ba5d0268da63eb06?s=32&d=identicon&r=PG](http://stackoverflow.com/users/3361233/user3361233)](http://stackoverflow.com/users/3361233/user3361233)  [user3361233](http://stackoverflow.com/users/3361233/user3361233)  211 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Take a look at my answer below, it doesn't rely on draft specifications. Your answer does allow the possibility to have different action urls, which mine doesn't. – [Tom Hofman](http://stackoverflow.com/users/545019/tom-hofman) [Apr 30 '14 at 12:10](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment35828861_22073362) |   add a comment |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 2 down vote | | I'm pretty late to the party, but here goes... My implementation borrows from @mkozicki but requires less hardcoded strings to get wrong. **Framework 4.5+ required**. Essentially, the controller method name should be the key to the routing.  **Markup**. The button name must be keyed with "action:[controllerMethodName]":  <form>  ... form fields ....  <button name="action:FundDeathStar" type="submit" formmethod="post">Fund Death Star</button>  <button name="action:HireBoba" type="submit" formmethod="post">Hire Boba Fett</button>  </form>  **Controller Methods**:  [SubmitActionToThisMethod]  public async Task<ActionResult> FundDeathStar(ImperialModel model)  {  await TrainStormTroopers();  return View();  }  [SubmitActionToThisMethod]  public async Task<ActionResult> HireBoba(ImperialModel model)  {  await RepairSlave1();  return View();  }  **Attribute Magic**. Notice the use of [CallerMemberName](https://msdn.microsoft.com/en-us/library/system.runtime.compilerservices.callermembernameattribute%28v=vs.110%29.aspx) goodness.  [AttributeUsage(AttributeTargets.Method, AllowMultiple = false, Inherited = true)]  public class SubmitActionToThisMethodAttribute : ActionNameSelectorAttribute  {  public SubmitActionToThisMethodAttribute([CallerMemberName]string ControllerMethodName = "")  {  controllerMethod = ControllerMethodName;  actionFormat = string.Concat(actionConstant, ":", controllerMethod);  }  const string actionConstant = "action";  readonly string actionFormat;  readonly string controllerMethod;  public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo)  {  var isValidName = false;  var value = controllerContext.Controller.ValueProvider.GetValue(actionFormat);  if (value != null)  {  controllerContext.Controller.ControllerContext.RouteData.Values[actionConstant] = controllerMethod;  isValidName = true;  }  return isValidName;  }  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/33663361)[improve this answer](http://stackoverflow.com/posts/33663361/edit) | [edited Nov 12 '15 at 18:31](http://stackoverflow.com/posts/33663361/revisions) | answered Nov 12 '15 at 2:22  [[https://i.stack.imgur.com/3vCE6.jpg?s=32&g=1](http://stackoverflow.com/users/459102/ajhuddy)](http://stackoverflow.com/users/459102/ajhuddy)  [ajhuddy](http://stackoverflow.com/users/459102/ajhuddy)  306315 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | Although this is a nice approach, you won't be able to use the built in mvc model binder because it uses the buttons "name" attribute. – [ooXei1sh](http://stackoverflow.com/users/1814739/ooxei1sh) [Jun 17 at 21:15](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment63235494_33663361) | | |  |  | | --- | --- | |  |  | | The purpose of this solution is a work around to the MVC post routing. Please describe an improvement. – [ajhuddy](http://stackoverflow.com/users/459102/ajhuddy" \o "306 reputation) [Jun 18 at 4:03](http://stackoverflow.com/questions/442704/how-do-you-handle-multiple-submit-buttons-in-asp-net-mvc-framework#comment63240426_33663361) |   add a comment |
| up vote 1 down vote | My JQuery approach using an extension method:  public static MvcHtmlString SubmitButtonFor<TController>(this HtmlHelper helper, Expression<Action<TController>> action, string value) where TController : Controller  {  RouteValueDictionary routingValues = Microsoft.Web.Mvc.Internal.ExpressionHelper.GetRouteValuesFromExpression(action);  var onclick = string.Format("$('form').first().attr('action', '/{0}')", string.Join("/", routingValues.Values.ToArray().Where(x => x != null).Select(x => x.ToString()).ToArray()));  var html = "<input type=\"submit\" value=\"" + value + "\" onclick=\"" + onclick + "\" />";  return MvcHtmlString.Create(html);  }  You can use it like this:  @(Html.SubmitButtonFor<FooController>(c => c.Save(null), "Save"))  And it renders like this:  <input type="submit" value="Save" onclick="$('form').first().attr('action', '/Foo/Save')" >   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/7823658)[improve this answer](http://stackoverflow.com/posts/7823658/edit) | answered Oct 19 '11 at 15:19  [[https://www.gravatar.com/avatar/aee677481611208883afaaeed5607060?s=32&d=identicon&r=PG](http://stackoverflow.com/users/356604/nahuelgq)](http://stackoverflow.com/users/356604/nahuelgq)  [NahuelGQ](http://stackoverflow.com/users/356604/nahuelgq)  786 | | |
|  | add a comment | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 1 down vote | | For each submit button just add:  $('#btnSelector').click(function () {  $('form').attr('action', "/Your/Action/);  $('form').submit();  });   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/11636095)[improve this answer](http://stackoverflow.com/posts/11636095/edit) | answered Jul 24 '12 at 17:23  [[https://i.stack.imgur.com/Izo4C.jpg?s=32&g=1](http://stackoverflow.com/users/752545/mishap)](http://stackoverflow.com/users/752545/mishap)  [mishap](http://stackoverflow.com/users/752545/mishap)  1,76183361 | |
|  | | add a comment |
| up vote 1 down vote | Based on mkozicki answer I come up with a bit different solution. I still use ActionNameSelectorAttribute But I needed to handle two buttons 'Save' and 'Sync'. They do almost the same so I didn't want to have two actions.  **attribute**:  public class MultipleButtonActionAttribute : ActionNameSelectorAttribute  {  private readonly List<string> AcceptedButtonNames;  public MultipleButtonActionAttribute(params string[] acceptedButtonNames)  {  AcceptedButtonNames = acceptedButtonNames.ToList();  }  public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo)  {  foreach (var acceptedButtonName in AcceptedButtonNames)  {  var button = controllerContext.Controller.ValueProvider.GetValue(acceptedButtonName);  if (button == null)  {  continue;  }  controllerContext.Controller.ControllerContext.RouteData.Values.Add("ButtonName", acceptedButtonName);  return true;  }  return false;  }  }  **view**  <input type="submit" value="Save" name="Save" />  <input type="submit" value="Save and Sync" name="Sync" />  **controller**  [MultipleButtonAction("Save", "Sync")]  public ActionResult Sync(OrgSynchronizationEditModel model)  {  var btn = this.RouteData.Values["ButtonName"];  I also want to point out that if actions do different things I would probably follow mkozicki post.   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/19020112)[improve this answer](http://stackoverflow.com/posts/19020112/edit) | answered Sep 26 '13 at 5:35  [[https://www.gravatar.com/avatar/ccf5c41a9b965314edffce685d4fc037?s=32&d=identicon&r=PG](http://stackoverflow.com/users/841898/petr-j)](http://stackoverflow.com/users/841898/petr-j)  [Petr J](http://stackoverflow.com/users/841898/petr-j)  558 | | | |
|  | add a comment | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 0 down vote | | When using ajax forms, we can use ActionLinks with POST HttpMethod and serialize the form in the AjaxOptions.OnBegin event.  Let's say you have two actions, InsertAction and UpdateAction:  <form>  @Html.Hidden("SomeField", "SomeValue")  @Ajax.ActionLink(  "Insert",  "InsertAction",  null,  new AjaxOptions {  OnBegin = "OnBegin",  UpdateTargetId = "yourDiv",  HttpMethod = "POST" })  @Ajax.ActionLink(  "Update",  "UpdateAction",  null,  new AjaxOptions {  OnBegin = "OnBegin",  UpdateTargetId = "yourDiv",  HttpMethod = "POST" })  </form>  Javascript  function OnBegin(xhr, settings) {  settings.data = $("form").serialize();  }   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/30107239)[improve this answer](http://stackoverflow.com/posts/30107239/edit) | [edited May 7 '15 at 18:42](http://stackoverflow.com/posts/30107239/revisions) | answered May 7 '15 at 16:50  [[https://www.gravatar.com/avatar/e9409d54f05964e755a18f63f88403d1?s=32&d=identicon&r=PG&f=1](http://stackoverflow.com/users/3616841/jbelanger)](http://stackoverflow.com/users/3616841/jbelanger)  [jBelanger](http://stackoverflow.com/users/3616841/jbelanger)  14114 | |
|  | | add a comment |
| up vote 0 down vote | I've created an **ActionButton** method for the **HtmlHelper**. It will generate normal input button with a bit of **javascript** in the **OnClick event** that will submit the form to the specified Controller/Action.  You use the helper like that  @Html.ActionButton("MyControllerName", "MyActionName", "button text")  this will generate the following HTML  <input type="button" value="button text" onclick="this.form.action = '/MyWebsiteFolder/MyControllerName/MyActionName'; this.form.submit();">  Here is the extension method code:  **VB.Net**  <System.Runtime.CompilerServices.Extension()>  Function ActionButton(pHtml As HtmlHelper, pAction As String, pController As String, pRouteValues As Object, pBtnValue As String, pBtnName As String, pBtnID As String) As MvcHtmlString  Dim urlHelperForActionLink As UrlHelper  Dim btnTagBuilder As TagBuilder  Dim actionLink As String  Dim onClickEventJavascript As String  urlHelperForActionLink = New UrlHelper(pHtml.ViewContext.RequestContext)  If pController <> "" Then  actionLink = urlHelperForActionLink.Action(pAction, pController, pRouteValues)  Else  actionLink = urlHelperForActionLink.Action(pAction, pRouteValues)  End If  onClickEventJavascript = "this.form.action = '" & actionLink & "'; this.form.submit();"  btnTagBuilder = New TagBuilder("input")  btnTagBuilder.MergeAttribute("type", "button")  btnTagBuilder.MergeAttribute("onClick", onClickEventJavascript)  If pBtnValue <> "" Then btnTagBuilder.MergeAttribute("value", pBtnValue)  If pBtnName <> "" Then btnTagBuilder.MergeAttribute("name", pBtnName)  If pBtnID <> "" Then btnTagBuilder.MergeAttribute("id", pBtnID)  Return MvcHtmlString.Create(btnTagBuilder.ToString(TagRenderMode.Normal))  End Function  **C#** (the C# code is just decompiled from the VB DLL, so it can get some beautification... but time is so short :-))  public static MvcHtmlString ActionButton(this HtmlHelper pHtml, string pAction, string pController, object pRouteValues, string pBtnValue, string pBtnName, string pBtnID)  {  UrlHelper urlHelperForActionLink = new UrlHelper(pHtml.ViewContext.RequestContext);  bool flag = Operators.CompareString(pController, "", true) != 0;  string actionLink;  if (flag)  {  actionLink = urlHelperForActionLink.Action(pAction, pController, System.Runtime.CompilerServices.RuntimeHelpers.GetObjectValue(pRouteValues));  }  else  {  actionLink = urlHelperForActionLink.Action(pAction, System.Runtime.CompilerServices.RuntimeHelpers.GetObjectValue(pRouteValues));  }  string onClickEventJavascript = "this.form.action = '" + actionLink + "'; this.form.submit();";  TagBuilder btnTagBuilder = new TagBuilder("input");  btnTagBuilder.MergeAttribute("type", "button");  btnTagBuilder.MergeAttribute("onClick", onClickEventJavascript);  flag = (Operators.CompareString(pBtnValue, "", true) != 0);  if (flag)  {  btnTagBuilder.MergeAttribute("value", pBtnValue);  }  flag = (Operators.CompareString(pBtnName, "", true) != 0);  if (flag)  {  btnTagBuilder.MergeAttribute("name", pBtnName);  }  flag = (Operators.CompareString(pBtnID, "", true) != 0);  if (flag)  {  btnTagBuilder.MergeAttribute("id", pBtnID);  }  return MvcHtmlString.Create(btnTagBuilder.ToString(TagRenderMode.Normal));  }  These methods have various parameters, but for the ease of use you can create some overload that take just the parameters you need.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/33168628)[improve this answer](http://stackoverflow.com/posts/33168628/edit) | [edited Oct 16 '15 at 12:42](http://stackoverflow.com/posts/33168628/revisions) | answered Oct 16 '15 at 10:48  [[https://www.gravatar.com/avatar/5f3f8f498f27e41002e07fddcbb5a19f?s=32&d=identicon&r=PG](http://stackoverflow.com/users/1057961/max)](http://stackoverflow.com/users/1057961/max)  [Max](http://stackoverflow.com/users/1057961/max)  3,90711325 | | |
|  | add a comment | |

|  |  |
| --- | --- |
| up vote 0 down vote | Use a custom helper(create a file "Helpers.cshtml" inside the App\_Code folder, at the root of your project) with javascript to rewrite (at an 'onclick' event) the form's 'action' attribute to something you want and then submit it.  The helper could be like:  @helper SubmitButton(string text, string controller,string action)  {  var uh = new System.Web.Mvc.UrlHelper(Context.Request.RequestContext);  string url = @uh.Action(action, controller, null);  <input type=button onclick="(  function(e)  {  $(e).parent().attr('action', '@url'); //rewrite action url  //create a submit button to be clicked and removed, so that onsubmit is triggered  var form = document.getElementById($(e).parent().attr('id'));  var button = form.ownerDocument.createElement('input');  button.style.display = 'none';  button.type = 'submit';  form.appendChild(button).click();  form.removeChild(button);  }  )(this)" value="@text"/>  }  And then use it as:  @Helpers.SubmitButton("Text for 1st button","ControllerForButton1","ActionForButton1")  @Helpers.SubmitButton("Text for 2nd button","ControllerForButton2","ActionForButton2")  ...  Inside your form. |