


Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.


Tell me more ×

How to handle checkboxes in ASP.NET MVC forms?



Appivate.ms
stackoverflow

Sign up for App Builder to receive tips/tricks for porting your iOS/Android app to Windows 8

sponsored by 

▲ This seems a bit bizarre to me, but as far as I can tell, this is how you do it.

192

▼ I have a collection of objects, and I want users to select one or more of them. This says to me "form with checkboxes." My objects don't have any concept of "selected" (they're rudimentary POCO's formed by deserializing a wcf call). So, I do the following:

★
100

```
public class SampleObject{
    public Guid Id {get;set;}
    public string Name {get;set;}
}
```

In the view:

```
<%
    using (Html.BeginForm())
    {
%>
<%foreach (var o in ViewData.Model) {%>
    <%=Html.CheckBox(o.Id)%>&nbsp;<%= o.Name %>
<%}%>
    <input type="submit" value="Submit" />
<%}%>
```

And, in the controller, this is the only way I can see to figure out what objects the user checked:

```
public ActionResult ThisLooksWeird(FormCollection result)
{
    var winners = from x in result.AllKeys
        where result[x] != "false"
        select x;
    // yadda
}
```

Its freaky in the first place, and secondly, for those items the user checked, the FormCollection lists its value as "true false" rather than just true.

Obviously, I'm missing something. I think this is built with the idea in mind that the objects in the collection that are acted upon within the html form are updated using `UpdateModel()` or through a `ModelBinder`.

But my objects aren't set up for this; does that mean that this is the only way? Is there another way to do it?

asp.net-mvc forms checkbox

share | improve this question

asked Oct 20 '08 at 21:30



Will ♦
64.1k ● 17 ● 143 ● 228

1 Others may find this solution useful: stackoverflow.com/questions/3291501/... – Aaron Jul 21 '10 at 11:51

tagged

asp.net-mvc × 59576

forms × 30205

checkbox × 6052

asked 4 years ago
viewed 128242 times
active 1 month ago

Community Bulletin

event **Port an app and you could win a trip to Microsoft's Build Conference**
— now through June 7th

CAREERS 2.0

Full Stack Django Engineer
SeatMe
San Francisco, CA/ relocation

Senior System Engineer (m/f)
Project A Ventures
Los Angeles, CA, Vereinigte...

Tracking Engineer
The New York Times
New York, NY

DATABASE ENGINEER (m/f) IN BERLIN
Zalando GmbH
Berlin, Germany / relocation

Infinite monkeys impractical - seeking talented developers...
Austin Digital, Inc. (GE...
Austin, TX

Data Visualization, Software Engineer at Sencha!
Sencha
Redwood City, CA/ remote /...

Linked

4 [How to maintain state of Html.CheckBox\(\) in ASP.NET MVC](#)

6 [Maintain state of a dynamic list of checkboxes in ASP.NET MVC](#)

4 [ASP.Net MVC - Handle Multiple Checkboxes](#)

6 [Getting all selected checkboxes from a FormCollection](#)

1 [MVC2 Checkbox problem](#)

1 [ASP.NET MVC Strongly](#)

20 Answers

active

oldest

votes

212

▲ Html.CheckBox is doing something weird - if you view source on the resulting page, you'll see there's an `<input type="hidden" />` being generated alongside each checkbox, which explains the "true false" values you're seeing for each form element.



Try this, which definitely works on ASP.NET MVC Beta because I've just tried it.

Put this in the view instead of using `Html.CheckBox()`:

```
<% using (Html.BeginForm("ShowData", "Home")) { %>
<% foreach (var o in ViewData.Model) { %>
    <input type="checkbox" name="selectedObjects" value="<%=o.Id%>">
    <%= o.Name %>
<%}%>
<input type="submit" value="Submit" />
<%}%>
```

Your checkboxes are all called `selectedObjects`, and the `value` of each checkbox is the GUID of the corresponding object.

Then post to the following controller action (or something similar that does something useful instead of `Response.Write()`)

```
public ActionResult ShowData(Guid[] selectedObjects) {
    foreach (Guid guid in selectedObjects) {
        Response.Write(guid.ToString());
    }
    Response.End();
    return (new EmptyResult());
}
```

This example will just write the GUIDs of the boxes you checked; ASP.NET MVC maps the GUID values of the selected checkboxes into the `Guid[] selectedObjects` parameter for you, and even parses the strings from the `Request.Form` collection into instantiated GUID objects, which I think is rather nice.

share | improve this answer

answered Oct 20 '08 at 21:49



Dylan Beattie

16.9k ●6●46●103

Yup! This is what I have had to do for my applications too! – [Adhip Gupta](#) Oct 21 '08 at 0:14

56 wtf. hidden input fields with the SAME name as the control. its ViewState 2.0! – [Simon Weaver](#) Jan 26 '09 at 9:20

2 This is also present in the 1.0 release. Look at the answer @andrea-balducci submitted that gives you an intelligent way to deal with this. If the checkbox is not checked, the resulting text retrieved should be 'false' - its a good workaround to account for brain dead browsers... – [Bryan Rehbein](#) Mar 27 '09 at 20:03

59 Surprise? Wtf? The hidden input has the same name as the checkbox - if the checkbox by the same name isn't checked then its value isn't posted whereas the value of the hidden is posted. The first time the browser encounters a named element it will use that value and ignore all other elements with the same name. This guarantees that a value is submitted: true if the checkbox is checked (assuming it's found above the hidden element) and false if checkbox is unchecked (the empty checkbox is ignored and the hidden becomes the fall back). The real wtf is why nobody else pointed this out. – [nerraga](#) Aug 17 '09 at 20:33

13 @nerraga: You're wrong: it's valid html to have multiple form elements with the same name; and if so, *all* elements are posted, and I'm not sure the order is actually defined (though likely to be simply in page order in practice). Using the word "false" as the value is somewhat misleading, so yeah, it's a WTF - a better, less misleading choice would have been something like "exists" or some meaningless token like a dash. – [Eamon Nerbonne](#) Dec 8 '10 at 17:27

show 3 more comments

CAREERS 2.0
by stackoverflow



Easily apply for your dream job
no formatting needed!

86 **HtmlHelper** adds an hidden input to notify the controller about Unchecked status. So to have the correct checked status:

```
bool bChecked = form[key].Contains("true");
```

share | improve this answer

answered Jan 12 '09 at 15:55



Andrea Balducci

1,823 ●8●16

This is the easiest approach to this problem and it's what I always use. It allows you to use the helper methods and account for ASP.NET MVCs behavior. – [Nick Daniels](#) Nov 19 '10 at 14:41

5 .. or in the case for not checked: `bool bChecked = form[key].Equals("false");` Doing a `.Contains("false")` fails as the true value is "true,false". – [Mark Robinson](#) Jan 24 '11 at 11:25

Typed View for class that has a collection of objects

- 2 The string was not identified as a valid Boolean string
- 2 knockout checkboxlist with dynamic data
- 0 CheckboxArray in MVC Framework
- 0 Form Submits same checkbox values on subsequent submit action in MVC

see more linked questions...

Related

- 0 Form Submits same checkbox values on subsequent submit action in MVC
- 854 How do I check a checkbox with jQuery or JavaScript?
- 899 Check checkbox checked property using jQuery
- 4 ASP.Net MVC - Handle Multiple Checkboxes
- 1 ASP.NET MVC form won't post
- 20 How to handle Booleans/CheckBoxes in ASP.NET MVC 2 with DataAnnotations?
- 6 Maintain state of a dynamic list of checkboxes in ASP.NET MVC
- 3 ASP.NET MVC form handling unknown number of inputs
- 1 How can I get multiple selections from ASP.NET CheckBoxes
- 2 ASP.net MVC 3 handling multiple checkboxes in each row of my table

44 In case you're wondering WHY they put a hidden field in with the same name as the checkbox the reason is as follows :

Comment from the sourcecode
MVCBetaSource\MVC\src\MvcFutures\Mvc\ButtonsAndLinkExtensions.cs

Render an additional `<input type="hidden".../>` for checkboxes. This addresses scenarios where unchecked checkboxes are not sent in the request. Sending a hidden input makes it possible to know that the checkbox was present on the page when the request was submitted.

I guess behind the scenes they need to know this for binding to parameters on the controller action methods. You could then have a tri-state boolean I suppose (bound to a nullable bool parameter). I've not tried it but I'm hoping that's what they did.

share | improve this answer

edited Jan 30 '09 at 0:05

answered Jan 26 '09 at 9:29

 [Simon Weaver](#)
22.4k ● 27 ● 185 ● 288

4 Yeah this is handy in scenarios where you have paged grids etc and you want to unselect an item that was previously selected in some business object. – [RichardOD](#) Jul 31 '09 at 14:45

41 You should also use `<label for="checkbox1">Checkbox 1</label>` because then people can click on the label text as well as the checkbox itself. It's also easier to style and at least in IE it will be highlighted when you tab through the page's controls.

```
<%= Html.CheckBox("cbNewColors", true) %><label for="cbNewColors">New colors</label>
```

This is not just a 'oh I could do it' thing. It's a significant user experience enhancement. Even if not all users know they can click on the label many will.

share | improve this answer

edited Mar 27 '09 at 21:14

answered Jan 26 '09 at 9:19

 [Simon Weaver](#)
22.4k ● 27 ● 185 ● 288

18 Here's what I've been doing.

View:

```
<input type="checkbox" name="applyChanges" />
```

Controller:

```
var checkBox = Request.Form["applyChanges"];

if (checkBox == "on")
{
    ...
}
```

I found the `Html.*` helper methods not so useful in some cases, and that I was better off doing it in plain old HTML. This being one of them, the other one that comes to mind is radio buttons.

Edit: this is on Preview 5, obviously YMMV between versions.

share | improve this answer

edited Oct 20 '08 at 21:59

answered Oct 20 '08 at 21:37

 [mmacaulay](#)
1,678 ● 8 ● 22

I just want to add an alternate example: `Object.Property != String.IsNullOrEmpty(Request.Form["NAME"]);` – [Alex](#) May 5 '10 at 0:26

if unchecked then it goes to exception – [Xulfee](#) Feb 28 '11 at 10:17

I'm surprised none of these answers used the built in MVC features for this.

11

I wrote a blog post about this [here](#), which even actually links the labels to the checkbox. I used the *EditorTemplate* folder to accomplish this in a clean and modular way.

You will simply end up with a new file in the *EditorTemplate* folder that looks like this:

```
@model SampleObject

@Html.CheckBoxFor(m => m.IsChecked)
@Html.HiddenFor(m => m.Id)
@Html.LabelFor(m => m.IsChecked, Model.Id)
```

in your actual view, there will be no need to loop this, simply 1 line of code:

```
@Html.EditorFor(x => ViewData.Model)
```

Visit my [blog post](#) for more details.

share | improve this answer

answered Oct 16 '11 at 3:31



Shawn Mclean

12.2k • 15 • 104 • 230

1 Thank you! Everbody still using the old forms ways. And when you submit you can access the Model without all this collection[] and request.form garbage - This is what I was looking for. +1 and +beer – [ppumkin](#) Mar 11 at 14:11

9

They appear to be opting to read the first value only, so this is "true" when the checkbox is checked, and "false" when only the hidden value is included. This is easily fetched with code like this:

```
model.Property = collection["ElementId"].ToLower().StartsWith("true");
```

share | improve this answer

answered Mar 20 '10 at 20:36



Fluffy

91 • 1 • 1

6

@Dylan Beattie Great Find!!! I Thank you much. To expand even further, this technique also works perfect with the View Model approach. MVC is so cool, it's smart enough to bind an array of Guids to a property by the same name of the Model object bound to the View. Example:

ViewModel:

```
public class SampleViewModel
{
    public IList<SampleObject> SampleObjectList { get; set; }
    public Guid[] SelectedObjectIds { get; set; }

    public class SampleObject
    {
        public Guid Id { get; set; }
        public string Name { get; set; }
    }
}
```

View:

```
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
<h2>Sample View</h2>
<table>
    <thead>
        <tr>
            <th>Checked</th>
            <th>Object Name</th>
        </tr>
    </thead>
    <% using (Html.BeginForm()) %>
    <%{>
        <tbody>
            <% foreach (var item in Model.SampleObjectList)
            { %>
                <tr>
                    <td><input type="checkbox" name="SelectedObjectIds" value="<%= ite
                    <td><%= Html.Encode(item.Name)%></td>
                </tr>
            <% } %>
```

```

</tbody>
</table>
<input type="submit" value="Submit" />
<%}%>

```

Controller:

```

[AcceptVerbs(HttpVerbs.Get)]
public ActionResult SampleView(Guid id)
{
    //Object to pass any input objects to the View Model Builder
    BuilderIO viewModelBuilderInput = new BuilderIO();

    //The View Model Builder is a conglomerate of repositories and methods use
    SampleViewModel viewModel = sampleViewModelBuilder.Build(viewModelBuilderI

    return View("SampleView", viewModel);
}

[AcceptVerbs(HttpVerbs.Post)]
public ActionResult SampleView(SampleViewModel viewModel)
{
    // The array of Guids successfully bound to the SelectedObjectIds property
    return View();
}

```

Anyone familiar with the View Model philosophy will rejoice, this works like a Champ!

[share](#) | [improve this answer](#)

answered Jun 4 '10 at 15:24



Thanks, your SampleViewModel cleared up some confusion in the original answer. – [parliament](#) Dec 21 '12 at 22:04

4 I'd also like to point out that you can name each checkbox a different name, and have that name part of the actionresults parameters.

4

Example,

View:

```

<%= Html.CheckBox("Rs232CheckBox", false, new { @id = "rs232" })%>RS-232
<%= Html.CheckBox("Rs422CheckBox", false, new { @id = "rs422" })%>RS-422

```

Controller:

```

public ActionResult MyAction(bool Rs232CheckBox, bool Rs422CheckBox) {
    ...
}

```

The values from the view are passed to the action since the names are the same.

I know this solution isn't ideal for your project, but I thought I'd throw the idea out there.

[share](#) | [improve this answer](#)

answered Mar 22 '10 at 14:48



3 This issue is happening in the release 1.0 as well. Html.Checkbox() causes another hidden field to be added with the same name/id as of your original checkbox. And as I was trying loading up a checkbox array using document.GetElemententsByName(), you can guess how things were getting messed up. It's a bizarre.

3

[share](#) | [improve this answer](#)

answered Jan 18 '10 at 11:23



I know that this question was written when MVC3 wasn't out, but for anyone who comes to this question and are using MVC3, you may want the "correct" way to do this

3

and are using MVC3, you may want the correct way to do this.

While I think that doing the whole

```
Contains("true");
```

thing is great and clean, and works on all MVC versions, the problem is that it doesn't take culture into account (as if it really matters in the case of a bool).

The "correct" way to figure out the value of a bool, at least in MVC3, is to use the ValueProvider.

```
var value = (bool)ValueProvider.GetValue("key").ConvertTo(typeof(bool));
```

I do this in one of my client's sites when I edit permissions:

```
var allPermissionsBase = Request.Params.AllKeys.Where(x => x.Contains("permission_"));
var allPermissions = new List<KeyValuePair<int, bool>>();

foreach (var key in allPermissionsBase)
{
    // Try to parse the key as int
    int keyAsInt;
    int.TryParse(key.Replace("permission_", ""), out keyAsInt);

    // Try to get the value as bool
    var value = (bool)ValueProvider.GetValue(key).ConvertTo(typeof(bool));
}
```

Now, the beauty of this is you can use this with just about any simple type, and it will even be correct based on the Culture (think money, decimals, etc).

The ValueProvider is what is used when you form your Actions like this:

```
public ActionResult UpdatePermissions(bool permission_1, bool permission_2)
```

but when you are trying to dynamically build these lists and check the values, you will never know the Id at compile time, so you have to process them on the fly.

share | improve this answer

edited Oct 24 '12 at 6:34

answered Jun 14 '12 at 23:07



Dan VanWinkle

259 ●2 ●10

is there a reason you use Request.Params instead of adding a FormCollection parameter to the method? – [SwissCoder](#) Oct 23 '12 at 9:55

No reason, I don't really see any benefit to one way over the other. – [Dan VanWinkle](#) Oct 24 '12 at 6:52

1 Sorry, there is actually a reason. Just looked at my code, and I am using this one method for 5 different calls, and I didn't feel like passing FormCollection from each method. This was the easiest way to get the keys without needing to pass anything around. – [Dan VanWinkle](#) Oct 24 '12 at 7:07

yeah I am new to MVC. And I read somewhere that it's better in general to use the FormCollection instead of Request.Params, same as using a typed Model instead of the FormCollection (so actually it would be best to use a model and avoid using Request.Params in general). I noticed other things of the code are not used, like the allPermissions variable and the keyAsInt. Thanks for answering. – [SwissCoder](#) Oct 24 '12 at 8:05

2

From the Controller:

```
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Index(FormCollection fc)
{
    var s = fc["checkbox1"];

    if (s == "on")
    {
        string x = "Hi";
    }
}
```

share | improve this answer

answered Jun 21 '10 at 19:41



kalyan

2,033 ●3 ●10 ●15



this is what i did to loose the double values when using the Html.CheckBox(...

1

```
Replace("true,false","true").Split(',')
```

with 4 boxes checked, unchecked, unchecked, checked it turns true,false,false,false,true,false into true,false,false,true. just what i needed

[share](#) | [improve this answer](#)

answered Oct 22 '10 at 16:07



Jeroen

11 ● 1



The easiest way to do is so...

1

You set the name and value.

```
<input type="checkbox" name="selectedProducts" value="@item.ProductId"
/>@item.Name
```

Then on submitting grab the values of checkboxes and save in an int array. then the appropriate LinQ Function. That's it..

```
[HttpPost]
public ActionResult Checkbox(int[] selectedObjects)
{
    var selected = from x in selectedObjects
                   from y in db
                   where y.ObjectId == x
                   select y;

    return View(selected);
}
```

[share](#) | [improve this answer](#)

answered Jan 5 '12 at 16:10



kk-dev11

305 ● 2 ● 3 ● 15



From what I can gather, the model doesn't want to guess whether checked = true or false, I got around this by setting a value attribute on the checkbox element with jQuery before submitting the form like this:

1

```
$('#input[type="checkbox"]').each(function () {
    $(this).attr('value', $(this).is(':checked'));
});
```

This way, you don't need a hidden element just to store the value of the checkbox.

[share](#) | [improve this answer](#)

answered May 22 '12 at 21:32



BraveNewMath

560 ● 4 ● 11



How about something like this?

0

```
bool isChecked = false;
if (Boolean.TryParse(Request.Form.GetValues("chkHuman")[0], out isChecked) == false)
    ModelState.AddModelError("chkHuman", "Nice try.");
```

[share](#) | [improve this answer](#)

answered Nov 11 '10 at 20:49



KO.

687 ● 1 ● 11 ● 25



When using the checkbox HtmlHelper, I much prefer to work with the posted checkbox form data as an array. I don't really know why, I know the other methods work, but I think I just prefer to treat comma separated strings as an array as much as possible.

0

So doing a 'checked' or true test would be:

```
//looking for [true],[false]
bool isChecked = form.GetValues(key).Contains("true");
```

Doing a false check would be:


```
//looking for [false],[false] or [false]
bool isChecked = !form.GetValues(key).Contains("true");
```

The main difference is to use `GetValues` as this returns as an array.

share | improve this answer

answered Apr 20 '11 at 22:25



eyesnz
1,778 ●2●5●17

Just do this on `$(document).ready` :

```
$('#input:hidden').each(function(e1) {
    var that = $(this)[0];
    if(that.id.length < 1 ) {

        console.log(that.id);
        that.parentElement.removeChild(that);

    }
});
```

share | improve this answer

edited Aug 10 '11 at 18:19

answered Aug 10 '11 at 15:38



Stack 101
1,992 ●9●27



doronAv
1

2 Unfortunately, you might get strange results on the server side for people with JavaScript turned off (NoScript plug-in, old mobile phones, Lynx...) – Arlick Sep 15 '11 at 21:55

My solution is:

```
<input type="checkbox" id="IsNew-checkbox" checked="checked" />
<input type="hidden" id="IsNew" name="IsNew" value="true" />
<script language="javascript" type="text/javascript" >
    $('#IsNew-checkbox').click(function () {
        if ($('#IsNew-checkbox').is(':checked')) {
            $('#IsNew').val('true');
        } else {
            $('#IsNew').val('false');
        }
    });
</script>
```

More you can find here: <http://www.blog.mieten.pl/2010/12/asp-net-mvc-custom-checkbox-as-solution-of-string-was-not-recognized-as-a-valid-boolean/>

share | improve this answer

edited Dec 3 '12 at 15:15

answered Dec 29 '10 at 10:20



bensiu
4,503 ●10●20●51



pawlom84
9 ●1

You don't need to use any Javascript for this ... `var isChecked = formData[key].Contains("true");` – Jammer Apr 16 at 8:36

I had nearly the same Problem but the return Value of my Controller was blocked with other Values.

Found a simple Solution but it seems a bit rough.

Try to type `Viewbag.` in your Controller and now you give it a name like `Viewbag.Checkbool`

Now switch to the View and try this `@Viewbag.Checkbool` with this you will get the value out of the Controller.

My Controller Parameters look like this:

```
public ActionResult Anzeigen(int productid = 90, bool islive = true)
```

and my Checkbox will update like this:

```
<input id="isLive" type="checkbox" checked="@ViewBag.Value" ONCLICK="window.locati
```

share | improve this answer

answered Apr 17 at 8:15



treborian

Your Answer

B *I* | | |

Sign up or [log in](#)

Post as a guest

Sign up using Google

Sign up using Facebook

Sign up using Stack Exchange

Name

Email

Post Your Answer

By posting your answer, you agree to the [privacy policy](#) and [terms of service](#).

Not the answer you're looking for? Browse other questions tagged [asp.net-mvc](#) [forms](#) [checkbox](#) or [ask your own question](#).

question feed

[about](#) [faq](#) [badges](#) [blog](#) [chat](#) [data](#) [legal](#) [privacy policy](#) [jobs](#) [advertising info](#) [mobile](#) [contact us](#) [feedback](#)

TECHNOLOGY

Stack Overflow
Server Fault
Super User
Web Applications
Ask Ubuntu
Webmasters
Game Development
TeX - LaTeX

Programmers
Unix & Linux
Ask Different (Apple)
WordPress Answers
Geographic Information Systems
Electrical Engineering
Android Enthusiasts
IT Security

Database Administrators
Drupal Answers
SharePoint
User Experience
Mathematica
[more \(14\)](#)

LIFE / ARTS

Photography
Science Fiction & Fantasy
Seasoned Advice (cooking)
Home Improvement
[more \(13\)](#)

CULTURE / RECREATION

English Language & Usage
Skeptics
Mi Yodeya (Judaism)
Travel
Arqade (gaming)
Bicycles
Role-playing Games
[more \(22\)](#)

SCIENCE

Mathematics
Cross Validated (stats)
Theoretical Computer Science
Physics
[more \(7\)](#)

OTHER

Stack Apps
Meta Stack Overflow
Area 51
Stack Overflow Careers