

# asp.net MVC handling dependencies between input controls?

Asked 15 years, 8 months ago   Modified 15 years, 8 months ago

Viewed 589 times



2



I have an application that has two dependent dropdown lists, where if the user selects a value in list box A, it updates the available set of inputs in list box B. Such as make/model of a car. when the user selects the manufacturer, the list of models would update accordingly.

In winforms, this would just be handled in the autopost back event. What technique/approach should I take in asp.net MVC? is done through AJAX ? just trying to get up to speed on MVC and looking to build strategies to handle common tasks I am asked to handle at work.

I wanted to thank everyone who contributed to answering this post.

asp.net-mvc

postback

data-entry

Share

edited Mar 31, 2009 at 1:26

Improve this question

Follow

asked Mar 30, 2009 at 21:08



MikeJ

14.6k ● 22 ● 74 ● 89

I wonder if I should rename the question to something like value cascades between controls ? – MikeJ Mar 30, 2009 at 22:46

3 Answers

Sorted by:

Highest score (default)



8

I'd do this through ajax. If you have these controls:

```
<select id="makes" /> and <select id="models" />
```



then you can do this with jquery:

```
$(document).ready(function() {  
    $("#select#makes").change(function() {  
        var make = this.value;  
        $.getJSON('models/list?make=' + make, function(  
            //load 2nd dropdown with result  
        ))  
    });  
});
```

Then you'd just need an action on the ModelsController called List() that returns a JSON data structure.

Share Improve this answer

Follow

answered Mar 30, 2009 at 21:31



Ben Scheirman

40.9k ● 21 ● 103 ● 139



2

That you are using ASP.NET MVC is somewhat irrelevant. You basically have three options for this type of UI mechanic on a web page.



1. If the data in your lists is relatively small and infrequently changing, it can be easiest just to pre-load all possible data on the page in the initial request either in something like a javascript array or hidden elements in the page markup. When the value of Box A changes, javascript just replaces the contents of Box B with the appropriate data. This all happens without any requests back to the server which makes it very fast. But this method breaks down when the size of your data impacts the response time of the initial page load.
2. If the data in your lists is large or frequently changing (within the time frame a user would be on the page making a decision), the legacy method is to just have the page get reloaded with the new query arguments when the value of Box A changes. Code on the back-end adjusts the output based on the new arguments. Depending on how complex the rest of the rendering code in your page is, this can be expensive.
3. This is a variation on option 2 and is basically the answer Ben Scheirman gave regarding ajax. You're still loading the contents of Box B on demand when Box A changes. The difference is that you're only reloading the piece of the page that has changed rather than the entire page.

Personally, if your data is small enough to use option 1, that's probably what I would go with due to its performance. Otherwise go with option 3, particularly if you've already got other ajax related things implemented already. Option 2 is seems to be considered a legacy mechanic by many people these days.

Share Improve this answer

edited Mar 30, 2009 at 21:58

Follow

answered Mar 30, 2009 at 21:50



Ryan Bolger

1,295 ● 8 ● 20

---

How do you get #2 to happen in MVC? Is there some JS that you have to attach to the select tag to post/submit when it changes? Sorry if htis sounds n00bish, but I didnt have the benefit of web development before winforms. – MikeJ Mar 30, 2009 at 22:13

---

Assuming you've included the jQuery libraries in your header, you can use the first two lines in Ben's example to setup the change event handler for the first select box. How best to then reload the page depends on whether you're passing arguments with GET or POST. – Ryan Bolger Mar 30, 2009 at 22:47

---



1

Functionally it would work to place those two drop lists in their own partial view and then return just that when the value in ListA is selected. If that isn't feasible for layout purposes, then Ben's method above looks good.



Share Improve this answer

Follow

answered Mar 30, 2009 at 21:33



[Nick DeVore](#)

10.2k ● 3 ● 40 ● 41



---

could you elaborate? I am really green to MVC and trying to pick up anything. – [MikeJ](#) Mar 30, 2009 at 21:44

---

Partial views allow you to update just one area of the page. When a request comes from the view to the controller, you can return just a partial view as a JSON (or anything) result. You have to handle this in AJAX though. – [Nick DeVore](#) Mar 31, 2009 at 15:05

---