[**MVC JSON - JsonResult and jQuery**](http://geekswithblogs.net/michelotti/archive/2008/06/28/mvc-json---jsonresult-and-jquery.aspx)

The latest release of the MVC framework provides the JsonResult for Controller actions. I was surprised that I did not find a weatlh of examples for usage so I figured it shouldn't be too hard to get a decent example going. It turns out, it was even easier than I anticipated. I wanted to create an example where I would invoke an AJAX call to dynamically populate a dropdown list.

[jQuery](http://jquery.com/) has recently received a huge surge of interest of the ASP.NET MVC community so I've been getting up to speed on it myself. I am quickly turning into a true believer and I've been impressed not only with how few lines of javascript it takes me to get things done but also how (relatively) easy it was to learn. In addition to numerous examples on the [jQuery web site](http://jquery.com/) I also recommend [jQuery In Action](http://www.amazon.com/jQuery-Action-Bear-Bibeault/dp/1933988355/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1214620855&sr=8-1).

Getting an AJAX request with jQuery and JSON in MVC is actually pretty simple. The first step is to have a Controller action that returns a JsonResult:

1: public JsonResult GetStateList()

2: {

3: List<ListItem> list = new List<ListItem>() {

4: new ListItem() { Value = "1", Text = "VA" },

5: new ListItem() { Value = "2", Text = "MD" },

6: new ListItem() { Value = "3", Text = "DC" }

7: };

8: return this.Json(list);

9: }

Here I'm using the System.Web.Mvc.ListItem class (which is used as items in the SelectList) to simply return a contrived list of US states to populate a dropdown list. All I have to do is call the Json() method passing in my object and it will automatically be serialized to JSON. When the request is made the AJAX call can be seen in the [Web Development Helper](http://projects.nikhilk.net/WebDevHelper/) which is an invaluable tool to any AJAX development:

Notice the "application/json" in the request header and the textual representation of the JSON that is being returned from my JSON controller action.

The jQuery to make this happen is quite succinct:

1: <script type="text/javascript">

2: $(function() {

3: $('#btnFillList').click(function() {

4: $.getJSON("/Home/GetStateList", null, function(data) {

5: $("#stateList").fillSelect(data);

6: });

7: });

8:

9: $('#btnClearList').click(function() {

10: $("#stateList").clearSelect();

11: });

12: });

13: </script>

Notice on line #4 the use of the [$.getJSON()](http://docs.jquery.com/Ajax/jQuery.getJSON) jQuery method. Then I've simply written a re-usable jQuery method called fillSelect() which will work on the JSON format returned by the ListItem class returned by my controller action:

1: $.fn.clearSelect = function() {

2: return this.each(function() {

3: if (this.tagName == 'SELECT')

4: this.options.length = 0;

5: });

6: }

7:

8: $.fn.fillSelect = function(data) {

9: return this.clearSelect().each(function() {

10: if (this.tagName == 'SELECT') {

11: var dropdownList = this;

12: $.each(data, function(index, optionData) {

13: var option = new Option(optionData.Text, optionData.Value);

14:

15: if ($.browser.msie) {

16: dropdownList.add(option);

17: }

18: else {

19: dropdownList.add(option, null);

20: }

21: });

22: }

23: });

24: }

On line #13 above you'll see the strongly-typed "Text" and "Value" properties that I have the luxury of using because of the native JSON serialization into objects.

The complete code sample can be downloaded [here](https://code.msdn.microsoft.com/Release/ProjectReleases.aspx?ProjectName=michelotti&ReleaseId=1216). It's barely 50 lines of code in all.

* Share This Post:
* Short Url: <http://wblo.gs/W73>

posted on Saturday, June 28, 2008 7:35 AM

|  |
| --- |
|  |

**Feedback**

[**#**](#396714)**re: MVC JSON - JsonResult and jQuery 7/31/2008 3:41 PM Rony Braga**

Great example!  
  
The code would be more clear if you had used Dictionary<> instead of listItem.  
  
In addicition you could create a helper method to return HTML code to render the control.

[**#**](#396719)**re: MVC JSON - JsonResult and jQuery 7/31/2008 4:34 PM Steve**

Rony - Thanks for the response. You can't use a Dictionary because internally the Dictionary keeps it's own order based on the hashing algorithm. The ListItem keeps the correct order.  
  
For the helper method, the "fillSelect()" is the helper method. Unless you were talking about a server side method that renders the client side jQuery code. If so, that's an interesting idea. I guess this would have to take several parameters though: a control which initiates the ajax request, the controller, the action, and the target dropdown control to fill.

[**#**](#400602)**re: MVC JSON - JsonResult and jQuery 9/23/2008 10:55 AM pete**

Great example. It really shows off what can be done with Jquery, MVC and Json.   
  
In real life I believe that the Ajax and Json is not going to be used for inital load of the page. I think an ordinary View would be less code. Once the page is loaded if you have dynamic parts of your page then the Jquery-Ajax and Json are going to come into play. A search page where the critera was at the top and the resluts at the bottom would be a good case for Jquery-Ajax Jason. A photo gallery with star rating for each picture. As user press star JQuery-Ajax updates db via json. A shoping cart would be another canidate for JQuery-Ajax.

[**#**](#400713)**re: MVC JSON - JsonResult and jQuery 9/24/2008 1:26 PM Steve**

Pete - thanks for the comment and those are all good examples. However, I think we're going to see more and more uses for AJAX even on the initial page load. It's often said that a singe-page web app is the "holy grail" for a pure ajax developer. If you look at sites like PageFlakes (http://www.pageflakes.com/) you'll see that \*everything\* is AJAX including the initial load.

[**#**](#400743)**re: MVC JSON - JsonResult and jQuery 9/24/2008 10:07 PM Steve**

Tom - You can get JSON with the MicrosoftAjax.js by using the Sys.Net.WebRequest invoke Method - the documentation can be found here: http://www.asp.net/AJAX/Documentation/Live/ClientReference/Sys.Net/WebRequestClass/WebRequestInvokeMethod.aspx.  
  
Additionally, this post shows an example for doing this: http://weblogs.asp.net/stephenwalther/archive/2008/09/06/asp-net-mvc-tip-41-creating-cascading-dropdown-lists-with-ajax.aspx

[**#**](#414291)**re: MVC JSON - JsonResult and jQuery 11/14/2008 8:46 PM Justin**

Sweet stuff. Really liking the combination of jQuery, JSON, and MVC. I still can't believe it's coming from Microsoft because it's... timely and trendy.   
  
Thanks for the example - I didn't know about ListItem and was previously passing generic lists.

[**#**](#421822)**re: MVC JSON - JsonResult and jQuery 12/2/2008 5:39 PM Doug Wilson**

Great post! I found this immensely useful.  
  
You can also do this:  
  
public JsonResult GetCityList()  
{  
var list = from city in repository.Cities.GetAll()  
select new { Text = city.Name, Value = city.ID };  
return Json(list.ToList());  
}  
  
This works really well in the Action, particularly when your type is more complex than what is required to populate a drop down list.

[**#**](#431334)**re: MVC JSON - JsonResult and jQuery 2/15/2009 7:57 PM Tsinghua**

I find the answer I want. Thx.

[**#**](#435903)**re: MVC JSON - JsonResult and jQuery 3/4/2009 5:12 PM Laurentiu Macovei**

What about if we need to return a cross-domain JsonP?

[**#**](#438996)**re: MVC JSON - JsonResult and jQuery 3/9/2009 10:25 AM AndrewO**

Great Post. Imagine how responsive an application can be if it is pure AJAX with JQuery calls.  
AO

[**#**](#438998)**re: MVC JSON - JsonResult and jQuery 3/9/2009 10:30 AM Steve**

@Laurentiu Macovei:  
  
I've never attempted to use JsonP in the context of MVC returning JSON. In general, I would look at this as a solution for a single domain. If you're wanting to build a web service that is truly a stand-alone web service that could be consumed by other domains, then I'd probably go with WCF rather than MVC. Here is an example: http://msdn.microsoft.com/en-us/library/cc716898.aspx

[**#**](#463377)**re: MVC JSON - JsonResult and jQuery 4/15/2009 6:01 AM Binal Shah**

hi,  
MVC application + homecontroler.cs + create.aspx + in asp.net page, i m using some html and asp.net controls (button,gridview).Now i want visibilty of gridview control is visible false and when i m click of button the visibilty of gridview must be true.what code i have to use for this with java script.  
please help me out.........................

[**#**](#463378)**re: MVC JSON - JsonResult and jQuery 4/15/2009 6:02 AM Binal Shah**

hi,  
MVC application + in asp.net page, i m using some html and asp.net controls (button,gridview).Now i want visibilty of gridview control is visible false and when i m click of button the visibilty of gridview must be true.what code i have to use for this with java script.  
please help me out.........................

[**#**](#463553)**re: MVC JSON - JsonResult and jQuery 4/15/2009 8:54 AM Steve**

@Binal - You are not meant to use ASP.NET server controls (like the grid view) in an MVC application. You should ensure that your <form> tag does \*not\* have a runat="server" attribute in it.   
  
To accomplish what you're seeking in MVC, you can essentially just follow the example above. When you click the button, you make an AJAX call which gets HTML from a controller action (this HTML being your grid) and then populates a <div> tag. This is a different approach than simply hiding/showing a control. Here is another example (though it was based on an old version of MVC): http://geekswithblogs.net/michelotti/archive/2008/07/06/mvc-componentcontroller-vs.-viewusercontrol.aspx

[**#**](#472130)**re: MVC JSON - JsonResult and jQuery 5/8/2009 4:09 PM N**

My Comment

[**#**](#476045)**re: MVC JSON - JsonResult and jQuery 6/5/2009 11:05 AM steve**

Seems like the basic idea being explored here could be librarized along side the other HTML input controls and we could have a self-generating, self-documenting CRUD to serve as front end to the db.

[**#**](#477835)**re: MVC JSON - JsonResult and jQuery 6/22/2009 11:07 AM Chris**

Nice. Never thought it would be quite that easy!

[**#**](#478545)**re: MVC JSON - JsonResult and jQuery 6/28/2009 5:49 PM Anonymous**

dude, stop saying everything is "easy". I'm sure it took you some time to figure this all out. I hate posters who say it's all easy when it's not. Even if jQuery makes it "eaiser" you sill have to understand the ins and outs of jQuery and learn how to parse through, etc.

[**#**](#480550)**re: MVC JSON - JsonResult and jQuery 7/12/2009 10:14 PM Tech Freaks**

Nice article.  
  
You can check a similar simple example for using Jquery/JSON for autopopulate select at http://www.tech-freaks.in/Java-Programming/Javascripts/jquery-json.html

[**#**](#485448)**re: MVC JSON - JsonResult and jQuery 8/23/2009 6:59 AM consacepo**

I've never attempted to use JsonP in the context of MVC returning JSON. In general, I would look at this as a solution for a single domain. If you're wanting to build a web service that is truly a stand-alone web service that could be consumed by other domains, then I'd probably go with WCF rather than MVC. Here is an example: http://msdn.microsoft.com/en-us/library/cc716898.aspx  
  
[consacepo](http://consacepo.blogspot.com/)

[**#**](#493924)**re: MVC JSON - JsonResult and jQuery 11/12/2009 12:45 AM congcong**

Great example!  
  
I want to ask a question ，how can do if the DataSet instead of List<ListItem>

[**#**](#494204)**re: MVC JSON - JsonResult and jQuery 11/13/2009 10:44 AM Steve**

@congcong - a DataSet is not a good candidate for a structure to be serialized to Json. I would put the contents of the DataSet into a SelectList before sending across the wire.

[**#**](#496457)**re: MVC JSON - JsonResult and jQuery 11/30/2009 3:28 AM congcong**

thankyou for your suggestion

[**#**](#499191)**re: MVC JSON - JsonResult and jQuery 12/17/2009 2:52 PM jvm**

I have controller method  
test(string id, string temp, string day)  
{  
}  
my temp value can be microsoft and google and ipod, how should i call using getJSON  
I am using  
$getJSON (url, function(json){});  
where url = ['./controller/methodname'].join('')  
but it gives error. If i set temp = "microsft" only, it works. So long string it is breaking. what can be issue?  
ANy help?

[**#**](#527947)**re: MVC JSON - JsonResult and jQuery 7/13/2010 5:50 AM Serjik**

It was great, tnx

[**#**](#531771)**re: MVC JSON - JsonResult and jQuery 8/6/2010 1:15 AM Paresh Rathod**

Great Post.  
  
But I am facing a problem.  
  
When I click on, submit button, controller's method is called and return json result.  
  
But my callback functions are not fired or called.  
  
So I m unable to fill my dropdown list.  
  
View file code is as follows:  
  
<script language="javascript" type="text/javascript">  
$(document).ready(function() {  
  
$("#btnFillDDL").click(function() {  
  
$.getJSON("Home/GetStateList", null, function(data) {  
$("#stateList").fillSelect(data);  
});  
});  
  
  
$.fn.clearSelect = function() {  
alert('Hi');  
return this.each(function() {  
if (this.tagName == 'SELECT')  
this.option.length = 0;  
});  
}  
  
$.fn.fillSelect = function(data) {  
alert(this.tagName);  
return this.clearSelect().each(function() {  
  
if (this.tagName == "SELECT") {  
var dropdownList = this;  
  
$.each(data, function(index, optionData) {  
var option = new Option(optionData.Text, optionData.Value);  
alert(index);  
if ($.browser.msie) {  
dropdownList.add(option);  
}  
else {  
dropdownList.add(null);  
}  
  
});  
}  
});  
}  
})  
</script>  
  
<% using (Html.BeginForm())  
{ %>  
<h2>  
<%= Html.Encode(ViewData["Message"]) %></h2>  
<p>  
To learn more about ASP.NET MVC visit <http://asp.net/mvc>.  
</p>  
<p>  
<select id="stateList">  
</select>  
</p>  
<p>  
<input type="submit" id="btnFillDDL" value="Fill State" />  
</p>  
<% } %>  
  
Please help me.  
  
Thanks in advance

[**#**](#531841)**re: MVC JSON - JsonResult and jQuery 8/6/2010 7:28 AM Steve Michelotti**

@Paresh - Did you put the functions fillSelect() and clearSelect() in a separate \*.js file? You should do that given these would be re-usable functions that you'd use in multiple areas of your application. Then make sure that you've included that javascript file \*before\* your code:  
  
<script type="text/javascript" src="/scripts/jquery.fillSelect.js"></script>  
  
<script language="javascript" type="text/javascript">  
$(document).ready(function() {  
  
$("#btnFillDDL").click(function() {  
  
$.getJSON("Home/GetStateList", null, function(data) {  
$("#stateList").fillSelect(data);  
});  
});  
</script>  
  
Those functions have to be defined \*first\* before they are called. My guess is that might be what you're running into. Also, you can download the sample code from the link in the post above and ensure that is running successfully for you.

[**#**](#539690)**re: MVC JSON - JsonResult and jQuery 9/25/2010 4:29 AM Tenh**

Dear all,   
  
I do all the step that you mentioned in above but it's not work and i also download your soft codes too for testing but it's still not work, how can i do? can u tell me because i need to know it's very much.

[**#**](#550002)**re: MVC JSON - JsonResult and jQuery 11/30/2010 11:28 AM J K**

I found that ListItem does not exist anymore in the mvc namespace, but this post was helpful: http://stackoverflow.com/questions/494156/does-system-web-mvc-listitem-no-longer-exist-in-asp-net-mvc-rc

[**#**](#563709)**re: MVC JSON - JsonResult and jQuery 2/19/2011 4:02 PM Biruk**

Great post. Worked like charm!

[**#**](#564983)**re: MVC JSON - JsonResult and jQuery 2/26/2011 5:15 PM PKS**

I am bit confuse. Is this a AJAX call? Because for AJAX call in jquery we should use $.ajax is it?

[**#**](#572517)**re: MVC JSON - JsonResult and jQuery 4/6/2011 9:13 AM Derek Morrison**

I noticed the Selected field in the JSON data. Did you omit this in the other code?

[**#**](#572691)**re: MVC JSON - JsonResult and jQuery 4/7/2011 6:27 AM Steve**

@Derek - That post is very old and was using a beta release of MVC back in summery 2008. Use SelectListItem instead of ListItem.

[**#**](#599440)**re: MVC JSON - JsonResult and jQuery 11/9/2011 4:36 PM sarsnake**

Your example, while nicely laid out, doesn't work for me. I replaced the ListItem with SelectedListItem, Json is still not valild and getJSON doesn't fire....please update this tutorial to avoid confusing people. thanks!

[A Simple JSON Model Binder >>](http://lozanotek.com/blog/archive/2010/04/16/simple_json_model_binder.aspx)

[**POSTing JSON Data to MVC Controllers**](http://lozanotek.com/blog/archive/2010/04/16/posting_json_data_to_mvc_controllers.aspx)

I’m sure that some (if not many) of you know this *little* approach for posting data from a [JSON](http://en.wikipedia.org/wiki/JSON) [POST](http://en.wikipedia.org/wiki/POST_(HTTP)) action (through jQuery) to an MSMVC Controller. If you don’t, keep reading otherwise go read something more important :)

**Note:**

As Phil noted, what the title should read is **Receiving JSON Data from An Action Method**, so my apologies for being misleading on the title. To see how you can actually post JSON data, check out Phil's [**Sending JSON to an ASP.NET MVC Action Method Argument**](http://haacked.com/archive/2010/04/15/sending-json-to-an-asp-net-mvc-action-method-argument.aspx) post.

**What are you trying to do?**

While working on MSMVC sites, I (like many of you) find myself doing a lot of user interaction with AJAX (via jQuery) on the client. At times, I will need to post several pieces of related data to an action and then get a response back from the server. If you’re doing some clever routing to create a more [RESTful](http://en.wikipedia.org/wiki/Representational_State_Transfer) resource, then this will not work for you since you’ll need to pass the information as dictated by your route. But if you have a controller action that takes a chunk of data, then this will work perfectly for you.

Again, the goal here is to keep things pretty simple and leverage as much as the runtime’s out of box wiring as possible.

**How do we do it?**

Take the following form (design and html):

<!-- Created via Html.EditorForModel extension method -->

<div class="editor-label">

<label for="Name">Name</label></div>

<div class="editor-field">

<input class="text-box single-line" id="Name" name="Name" type="text" value="" /> </div>

<div class="editor-label">

<label for="Age">Age</label></div>

<div class="editor-field">

<input class="text-box single-line" id="Age" name="Age" type="text" value="" /> </div>

<p><input type="submit" value="Save" id="personCreate" /></p>

<div><span id="resultMessage"></span></div>

[view raw](https://gist.github.com/raw/367996/7ba399449edf2a8397d8e7adb259231d54642206/gistfile1.html)[gistfile1.html](#file_gistfile1.html)[This Gist](https://gist.github.com/367996) brought to you by [GitHub](http://github.com/).

The whole purpose of this form is to take the data, do some simple data validation and then POST the data to the server for processing. This can be done in several ways, since we’re doing a POST action, we opt to use the [jQuery.post()](http://api.jquery.com/jQuery.post/) method. As the documentation states, this is nothing more than an explicit setting of the [jQuery.ajax()](http://api.jquery.com/jQuery.ajax/) API function which specifies POST as the action method.

One of the parameters that can be passed to the post API call is the actual data for the call. The type of this data parameter can be either a map (structure) or string since this will be passed as part of the request body. To get the most “bang for your buck” with this API, you’ll want to have the data parameter be a map (structure) and let the jQuery and MSMVC runtime do the heavy lifting for you. Let’s examine the data that’s request by our controller for the above form.

All we need for this request to be valid is a nothing more than a simple input model to hold the “person” information as well as an action to receive it:

namespace JSONMvcSample.Models {

using System;

[Serializable]

public class PersonInputModel {

public string Name { get; set; }

public int Age { get; set; }

}

}

namespace JSONMvcSample.Controllers {

using System.Web.Mvc;

using Models;

[HandleError]

public class HomeController : Controller {

public ActionResult Index() {

return View();

}

[HttpPost]

public ActionResult Save(PersonInputModel inputModel) {

string message = string.Format("Created user '{0}' in the system.", inputModel.Name);

return Json(new PersonViewModel {Message = message});

}

}

}

[view raw](https://gist.github.com/raw/368004/ce847f858df334a0608b5b706d0c506275176b57/gistfile1.cs)[gistfile1.cs](#file_gistfile1.cs)[This Gist](https://gist.github.com/368004) brought to you by [GitHub](http://github.com/).

As you can see, there's nothing complicated about these pieces. Just a simple controller action that takes in an Input (View) Model and returns a JSON data structure back to the caller. The only thing that can be consider complicated about this action is the [[HttpPost]](http://msdn.microsoft.com/en-us/library/system.web.mvc.httppostattribute(VS.100).aspx) restriction, since as we stated earlier, we're doing a jQuery post API call.

One of the nicest features of MSMVC is the concept of a [IModelBinder](http://msdn.microsoft.com/en-us/library/system.web.mvc.imodelbinder.aspx), which performs a lot of heavy lifting for you when [you transfer complex objects via a request](http://www.hanselman.com/blog/IPrincipalUserModelBinderInASPNETMVCForEasierTesting.aspx). Out of the box, the [DefaultModelBinder](http://msdn.microsoft.com/en-us/library/system.web.mvc.defaultmodelbinder.aspx) will wire up primitive, (light) complex types and collections for you by pulling information from the request (request body, query string, form data, etc.) In other words, if you provide the right information for a request, the DefaultModelBinder will do the rest for you. Let’s go back and examine our jQuery POST code.

What we need to do here is simple,

* Get the values out of the input elements
* Perform (poor/simple) validation
* POST the data to the server
* Get the response and display the information

The previous list can be accomplished by the following:

$(function () {

$("#personCreate").click(function () {

var person = getPerson();

// poor man's validation

if (person == null) {

alert("Specify a name please!");

return;

}

// take the data and post it via json

$.post("home/save", person, function (data) {

// get the result and do some magic with it

var message = data.Message;

$("#resultMessage").html(message);

});

});

});

function getPerson() {

var name = $("#Name").val();

var age = $("#Age").val();

// poor man's validation

return (name == "") ? null : { Name: name, Age: age };

}

[view raw](https://gist.github.com/raw/368014/386689d970d61cbcf43e62f96a00a32d94e028eb/gistfile1.js)[gistfile1.js](#file_gistfile1.js)[This Gist](https://gist.github.com/368014) brought to you by [GitHub](http://github.com/).

In the above sample, all the magic takes place with the aide of the "Person" structure that's passed as a parameter to the POST call. If we examine the request body (with the aide of Firebug) we can see how the request is actually structured:

The outlined area show that the data within the request is passed as key-value pairs ([application/x-www-form-urlencoded](#h-17.13.4.1)) to the server processing. Again, since MSMVC comes with the DefaultModelBinder, we get the mapping of the request data into our input (view) model for free:

From here, we can easily return JSON back to the client and allow it to do processing as it sees fit. In our case, we’ll just display the data to the user as a simple message:

Pretty straight forward and simple, huh? If you’re interested on the code, go here: <http://github.com/jglozano/samples/tree/master/JSONMvc>

Happy Coding!

Friday, April 16, 2010 12:09 AM

**Your Comments.**