[Use ASP.NET MVC validation with jquery ajax?](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

[Ask Question](https://stackoverflow.com/questions/ask)

Asked 10 years, 11 months ago

Modified [5 years, 8 months ago](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax?lastactivity)

Viewed 162k times

Report this ad

**126**

I have simple ASP.NET MVC action like this :

public ActionResult Edit(EditPostViewModel data)

{

}

The EditPostViewModel have validation attributes like this :

[Display(Name = "...", Description = "...")]

[StringLength(100, MinimumLength = 3, ErrorMessage = "...")]

[Required()]

public string Title { get; set; }

In the view I am using the following helpers :

@Html.LabelFor(Model => Model.EditPostViewModel.Title, true)

@Html.TextBoxFor(Model => Model.EditPostViewModel.Title,

new { @class = "tb1", @Style = "width:400px;" })

If I do a submit on a form that this textbox is placed in a validation will be done first on client and then on service(ModelState.IsValid).

Now I got a couple of questions :

1. Can this be used with jQuery ajax submit instead? What I am doing is simply remove the form and on clicking the submit button a javascript will gather data and then run the $.ajax.
2. Will the server side ModelState.IsValid work?
3. How can I forward validation problem back to the client and present it as if Im using the build int validation(@Html.ValidationSummary(true))?

Example of Ajax call :

function SendPost(actionPath) {

$.ajax({

url: actionPath,

type: 'POST',

dataType: 'json',

data:

{

Text: $('#EditPostViewModel\_Text').val(),

Title: $('#EditPostViewModel\_Title').val()

},

success: function (data) {

alert('success');

},

error: function () {

alert('error');

}

});

}

Edit 1:

Included on page :

<script src="/Scripts/jquery-1.7.1.min.js"></script>

<script src="/Scripts/jquery.validate.min.js"></script>

<script src="/Scripts/jquery.validate.unobtrusive.min.js"></script>

* [jquery](https://stackoverflow.com/questions/tagged/jquery)
* [ajax](https://stackoverflow.com/questions/tagged/ajax)
* [asp.net-mvc](https://stackoverflow.com/questions/tagged/asp.net-mvc)
* [validation](https://stackoverflow.com/questions/tagged/validation)

[Share](https://stackoverflow.com/q/14005773)

[Improve this question](https://stackoverflow.com/posts/14005773/edit)

Follow

[edited Dec 28, 2012 at 10:33](https://stackoverflow.com/posts/14005773/revisions)

asked Dec 22, 2012 at 19:16

[[](https://stackoverflow.com/users/1490727/ivy)](https://stackoverflow.com/users/1490727/ivy)

[Ivy](https://stackoverflow.com/users/1490727/ivy)

**2,285**44 gold badges2424 silver badges2929 bronze badges

* Nice answer below. Here's a related question. The answer allows for client-side or server-side validation. I'm in love with the JQuery code they provide. (No, it wasn't my answer.) [stackoverflow.com/questions/28987752/…](http://stackoverflow.com/questions/28987752/jquery-post-and-unobtrusive-ajax-validation-not-working-mvc-4)

– [Adventure](https://stackoverflow.com/users/546599/adventure)

[Mar 9, 2017 at 23:38](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment72536920_14005773)

[Add a comment](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

5 Answers

Sorted by:



**160**

+50

**Client Side**

Using the jQuery.validate library should be pretty simple to set up.

Specify the following settings in your Web.config file:

<appSettings>

<add key="ClientValidationEnabled" value="true"/>

<add key="UnobtrusiveJavaScriptEnabled" value="true"/>

</appSettings>

When you build up your view, you would define things like this:

@Html.LabelFor(Model => Model.EditPostViewModel.Title, true)

@Html.TextBoxFor(Model => Model.EditPostViewModel.Title,

new { @class = "tb1", @Style = "width:400px;" })

@Html.ValidationMessageFor(Model => Model.EditPostViewModel.Title)

***NOTE:****These need to be defined within a form element*

Then you would need to include the following libraries:

<script src='@Url.Content("~/Scripts/jquery.validate.js")' type='text/javascript'></script>

<script src='@Url.Content("~/Scripts/jquery.validate.unobtrusive.js")' type='text/javascript'></script>

This should be able to set you up for client side validation

Resources

* <http://msdn.microsoft.com/en-us/vs2010trainingcourse_aspnetmvccustomvalidation_topic5.aspx>

**Server Side**

**NOTE:** This is only for additional server side validation on top of jQuery.validation library

Perhaps something like this could help:

[ValidateAjax]

public JsonResult Edit(EditPostViewModel data)

{

//Save data

return Json(new { Success = true } );

}

Where ValidateAjax is an attribute defined as:

public class ValidateAjaxAttribute : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

if (!filterContext.HttpContext.Request.IsAjaxRequest())

return;

var modelState = filterContext.Controller.ViewData.ModelState;

if (!modelState.IsValid)

{

var errorModel =

from x in modelState.Keys

where modelState[x].Errors.Count > 0

select new

{

key = x,

errors = modelState[x].Errors.

Select(y => y.ErrorMessage).

ToArray()

};

filterContext.Result = new JsonResult()

{

Data = errorModel

};

filterContext.HttpContext.Response.StatusCode =

(int) HttpStatusCode.BadRequest;

}

}

}

What this does is return a JSON object specifying all of your model errors.

Example response would be

[{

"key":"Name",

"errors":["The Name field is required."]

},

{

"key":"Description",

"errors":["The Description field is required."]

}]

This would be returned to your error handling callback of the $.ajax call

You can loop through the returned data to set the error messages as needed based on the Keys returned (I think something like $('input[name="' + err.key + '"]') would find your input element

[Share](https://stackoverflow.com/a/14106135)

[Improve this answer](https://stackoverflow.com/posts/14106135/edit)

Follow

[edited May 22, 2015 at 18:26](https://stackoverflow.com/posts/14106135/revisions)

answered Dec 31, 2012 at 21:20

[[](https://stackoverflow.com/users/12096/andrew-burgess)](https://stackoverflow.com/users/12096/andrew-burgess)

[Andrew Burgess](https://stackoverflow.com/users/12096/andrew-burgess)

**5,300**55 gold badges3131 silver badges3737 bronze badges

* 1

Great answer, especially with the awesome ValidateAjaxAttribute! Thank you!

– [René](https://stackoverflow.com/users/280392/ren%c3%a9)

[Feb 4, 2013 at 14:19](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment20537538_14106135)

* 4

I don't understand why this answer got so many votes. It doesn't answer question 1: how to do client validation when posting with $.ajax? I think @Shyju answer helps with that.

– [Valentin](https://stackoverflow.com/users/443379/valentin)

[Jul 22, 2014 at 12:10](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment38657112_14106135)

* 2

@Valentin - my answer does help though because the data is validated server side as well. The validation plugins should enable dynamic validation as the form is filled out, and when the form is submitted (however the OP wants to do that), the server will provide final validation, which is preferable anyways since client side validation can be bypassed.

– [Andrew Burgess](https://stackoverflow.com/users/12096/andrew-burgess)

[Jul 23, 2014 at 8:14](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment38691462_14106135)

* 10

I utilise jQuery's validation message spans by looping through the errors returned and inserting the error message into the correct span: for (var i = 0; i < modelStateErrors.length; i++) { $('span[data-valmsg-for="' + modelStateErrors[i].key + '"]').text(modelStateErrors[i].errors[0]); }

– [Ian](https://stackoverflow.com/users/2456110/ian)

[Oct 15, 2015 at 13:52](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment54110097_14106135)

* 10

This answer is great! But I still think the ASP.NET MVC framework should provide a built-in way of doing that.

– [Zignd](https://stackoverflow.com/users/1324082/zignd" \o "6,906 reputation)

[Nov 29, 2015 at 4:42](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment55715147_14106135)

[Show **1** more comment](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

**42**

What you should do is to serialize your form data and send it to the controller action. ASP.NET MVC will bind the form data to the EditPostViewModel object( your action method parameter), using MVC model binding feature.

You can validate your form at client side and if everything is fine, send the data to server. The [valid()](https://jqueryvalidation.org/valid/) method will come in handy.

$(function () {

$("#yourSubmitButtonID").click(function (e) {

e.preventDefault();

var \_this = $(this);

var \_form = \_this.closest("form");

var isvalid = \_form .valid(); // Tells whether the form is valid

if (isvalid)

{

$.post(\_form.attr("action"), \_form.serialize(), function (data) {

//check the result and do whatever you want

})

}

});

});

[Share](https://stackoverflow.com/a/14061592)

[Improve this answer](https://stackoverflow.com/posts/14061592/edit)

Follow

[edited Oct 11, 2017 at 17:56](https://stackoverflow.com/posts/14061592/revisions)

answered Dec 27, 2012 at 21:33

[[](https://stackoverflow.com/users/40521/shyju)](https://stackoverflow.com/users/40521/shyju)

[Shyju](https://stackoverflow.com/users/40521/shyju)

**215k**105105 gold badges414414 silver badges497497 bronze badges

* 1

Thanks! I tried this $("form#" + formId).validate() but it says that the form(that is found) do not have a validate()?

– [Ivy](https://stackoverflow.com/users/1490727/ivy)

[Dec 28, 2012 at 10:22](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment19446393_14061592)

* See Edit1 where I show that the validate script is included on the webpage. I also see that the validation is running when using a regular input type submit button.

– [Ivy](https://stackoverflow.com/users/1490727/ivy)

[Dec 28, 2012 at 10:34](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment19446626_14061592)

* I found the problem(removed Scripts.Render from masterpage). But still have problems to send back ModelState validation errors to the client? How am I supose to thandle that? For example if the user is not loggedin anymore( ModelState.AddModelError("CustomError", "validation text").

– [Ivy](https://stackoverflow.com/users/1490727/ivy)

[Dec 28, 2012 at 13:48](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment19450548_14061592)

* 1

you need to include the jquery.validate and jquery.validate.unobtrusive js files in your page. IS your HTML inputs have the attribute which validation plugins look for ?

– [Shyju](https://stackoverflow.com/users/40521/shyju)

[Dec 28, 2012 at 14:51](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment19451920_14061592)

* 1

Ivy : Your return type (ActionResult) is a base class of JsonResult. so it can return JSON data. What you should do is, If it is an ajax call (check Request.IsAjax), get the validation errors and build a JSON and send it back to client. Check the json in the callback method of $.post and show the error messages.

– [Shyju](https://stackoverflow.com/users/40521/shyju)

[Dec 29, 2012 at 23:52](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment19481506_14061592)

[Show **2** more comments](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

**10**

Here's a rather simple solution:

In the controller we return our errors like this:

if (!ModelState.IsValid)

{

return Json(new { success = false, errors = ModelState.Values.SelectMany(x => x.Errors).Select(x => x.ErrorMessage).ToList() }, JsonRequestBehavior.AllowGet);

}

Here's some of the client script:

function displayValidationErrors(errors)

{

var $ul = $('div.validation-summary-valid.text-danger > ul');

$ul.empty();

$.each(errors, function (idx, errorMessage) {

$ul.append('<li>' + errorMessage + '</li>');

});

}

That's how we handle it via ajax:

$.ajax({

cache: false,

async: true,

type: "POST",

url: form.attr('action'),

data: form.serialize(),

success: function (data) {

var isSuccessful = (data['success']);

if (isSuccessful) {

$('#partial-container-steps').html(data['view']);

initializePage();

}

else {

var errors = data['errors'];

displayValidationErrors(errors);

}

}

});

Also, I render partial views via ajax in the following way:

var view = this.RenderRazorViewToString(partialUrl, viewModel);

return Json(new { success = true, view }, JsonRequestBehavior.AllowGet);

RenderRazorViewToString method:

public string RenderRazorViewToString(string viewName, object model)

{

ViewData.Model = model;

using (var sw = new StringWriter())

{

var viewResult = ViewEngines.Engines.FindPartialView(ControllerContext,

viewName);

var viewContext = new ViewContext(ControllerContext, viewResult.View,

ViewData, TempData, sw);

viewResult.View.Render(viewContext, sw);

viewResult.ViewEngine.ReleaseView(ControllerContext, viewResult.View);

return sw.GetStringBuilder().ToString();

}

}

[Share](https://stackoverflow.com/a/37409393)

[Improve this answer](https://stackoverflow.com/posts/37409393/edit)

Follow

[edited Aug 30, 2017 at 10:41](https://stackoverflow.com/posts/37409393/revisions)

answered May 24, 2016 at 9:24

[[](https://stackoverflow.com/users/5374333/alex-herman)](https://stackoverflow.com/users/5374333/alex-herman)

[Alex Herman](https://stackoverflow.com/users/5374333/alex-herman)

**2,718**44 gold badges3434 silver badges5353 bronze badges

* 3

Your "GetModelStateErrors" can be simplified completely by converting it to a way simpler single-line statement: **return ModelState.Values.SelectMany(x => x.Errors).Select(x => x.ErrorMessage).ToList();**

– [Camilo Terevinto](https://stackoverflow.com/users/2141621/camilo-terevinto)

[Sep 3, 2016 at 0:44](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment65938518_37409393)

* 1

Why not simply return a PartialView to render to Ajax?

– [Sinjai](https://stackoverflow.com/users/5043056/sinjai" \o "1,165 reputation)

[Aug 24, 2017 at 20:18](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax#comment78699327_37409393)

[Add a comment](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

**5**

Added some more logic to solution provided by @Andrew Burgess. Here is the full solution:

Created a action filter to get errors for ajax request:

public class ValidateAjaxAttribute : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

if (!filterContext.HttpContext.Request.IsAjaxRequest())

return;

var modelState = filterContext.Controller.ViewData.ModelState;

if (!modelState.IsValid)

{

var errorModel =

from x in modelState.Keys

where modelState[x].Errors.Count > 0

select new

{

key = x,

errors = modelState[x].Errors.

Select(y => y.ErrorMessage).

ToArray()

};

filterContext.Result = new JsonResult()

{

Data = errorModel

};

filterContext.HttpContext.Response.StatusCode =

(int)HttpStatusCode.BadRequest;

}

}

}

Added the filter to my controller method as:

[HttpPost]

// this line is important

[ValidateAjax]

public ActionResult AddUpdateData(MyModel model)

{

return Json(new { status = (result == 1 ? true : false), message = message }, JsonRequestBehavior.AllowGet);

}

Added a common script for jquery validation:

function onAjaxFormError(data) {

var form = this;

var errorResponse = data.responseJSON;

$.each(errorResponse, function (index, value) {

// Element highlight

var element = $(form).find('#' + value.key);

element = element[0];

highLightError(element, 'input-validation-error');

// Error message

var validationMessageElement = $('span[data-valmsg-for="' + value.key + '"]');

validationMessageElement.removeClass('field-validation-valid');

validationMessageElement.addClass('field-validation-error');

validationMessageElement.text(value.errors[0]);

});

}

$.validator.setDefaults({

ignore: [],

highlight: highLightError,

unhighlight: unhighlightError

});

var highLightError = function(element, errorClass) {

element = $(element);

element.addClass(errorClass);

}

var unhighLightError = function(element, errorClass) {

element = $(element);

element.removeClass(errorClass);

}

Finally added the error javascript method to my Ajax Begin form:

@model My.Model.MyModel

@using (Ajax.BeginForm("AddUpdateData", "Home", new AjaxOptions { HttpMethod = "POST", OnFailure="onAjaxFormError" }))

{

}

[Share](https://stackoverflow.com/a/49797725)

[Improve this answer](https://stackoverflow.com/posts/49797725/edit)

Follow

answered Apr 12, 2018 at 13:21

[[](https://stackoverflow.com/users/5522140/manprit-singh-sahota)](https://stackoverflow.com/users/5522140/manprit-singh-sahota)

[Manprit Singh Sahota](https://stackoverflow.com/users/5522140/manprit-singh-sahota)

**1,299**22 gold badges1515 silver badges3939 bronze badges

[Add a comment](https://stackoverflow.com/questions/14005773/use-asp-net-mvc-validation-with-jquery-ajax)

**1**

You can do it this way:

(**Edit:** Considering that you're waiting for a response json with dataType: 'json')

.NET

public JsonResult Edit(EditPostViewModel data)

{

if(ModelState.IsValid)

{

// Save

return Json(new { Ok = true } );

}

return Json(new { Ok = false } );

}

JS:

success: function (data) {

if (data.Ok) {

alert('success');

}

else {

alert('problem');

}

},

If you need I can also explain how to do it by returning a error 500, and get the error in the event error (ajax). But in your case this may be an option