**Using AJAX to submit a form in Asp.Net MVC 4: A Paraphrased Guide**

Author: Guadalupe HornerDate: 2023-03-19

To implement validation rules in your project, start by including a file in your \_layout.cshtml. Next, initialize the validator outside of the $(document).ready() function in your js file/script. You may also add or remove more validation rules as desired. Once set up, validate the inputs using the provided HTML page example. Note that the updateUi function is a custom function that updates the error spans to show/hide. However, some users have reported difficulty displaying error messages during server-side validation. To troubleshoot this issue, review the \_Layout.cshtml, Index.cshtml, categoryType.js, Model: CategoryType.cs, and CategoryTypesController.cs files. It may be that an error message is not appearing due to incorrect implementation of the validation rules.

**Table of contents**

* [Submit form using AJAX in Asp.Net mvc 4](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#submit-form-using-ajax-in-aspnet-mvc-4)
* [Ajax Calls with Fetch API in .NET Core 3.1](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#ajax-calls-with-fetch-api-in-net-core-31)
* [Bootstrap modal+ajax+jquery+validation+.net core 2.0](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#bootstrap-modalajaxjqueryvalidationnet-core-20)
* [Is there an unobtrusive form in Ajax JS?](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#is-there-an-unobtrusive-form-in-ajax-js)
* [What is unobtrusive JavaScript?](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#what-is-unobtrusive-javascript)
* [How to insert jQuery Ajax form submit using serializing?](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#how-to-insert-jquery-ajax-form-submit-using-serializing)
* [How to use Ajax in ASP NET MVC?](https://copyprogramming.com/howto/submit-form-using-ajax-in-asp-net-mvc-4#how-to-use-ajax-in-asp-net-mvc)

Ezoic

**Submit form using AJAX in Asp.Net mvc 4**

**Question:**

As I delve into learning asp.net, I have successfully implemented the ability to load page contents without refreshing utilizing **Ajax.Actionlink** and **AjaxOptions()** . However, I am encountering difficulty in incorporating ajax when submitting a form. Despite extensive research, I have yet to discover a suitable solution. Below are the codes I have implemented thus far.

Controller page

namespace CrudMvc.Controllers

{

public class HomeController : Controller

{

sampleDBEntities db = new sampleDBEntities();

//

// GET: /Home/

public ActionResult Index()

{

return View(db.myTables.ToList());

}

public PartialViewResult Details(int id = 0)

{

myTable Table = db.myTables.Find(id);

return PartialView(Table);

}

[HttpGet]

public PartialViewResult Create()

{

return PartialView();

}

[HttpPost]

public ActionResult Create(myTable table)

{

if (ModelState.IsValid)

{

db.myTables.Add(table);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(table);

}

protected override void Dispose(bool disposing)

{

db.Dispose();

base.Dispose(disposing);

}

}

}

**Index** View page

Ezoic

@model IEnumerable

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

**Index**

@Ajax.ActionLink("Add New", "Create", new AjaxOptions()

{

HttpMethod = "GET",

UpdateTargetId = "info",

InsertionMode = InsertionMode.Replace

})

@foreach (var item in Model) {

}

|  |  |
| --- | --- |
| **@Html.DisplayNameFor(model => model.name)** | **Action** |
| @Html.DisplayFor(modelItem => item.name) | @Ajax.ActionLink("Details", "Details", new{ id=item.id}, new AjaxOptions() { HttpMethod = "GET", UpdateTargetId = "info", InsertionMode = InsertionMode.Replace }) |

**Create** View Page

Ezoic

@model CrudMvc.Models.myTable

@{

ViewBag.Title = "Create";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

**Create**

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)

myTable

@Html.LabelFor(model => model.id)

Ezoic

@Html.EditorFor(model => model.id)

@Html.ValidationMessageFor(model => model.id)

@Html.LabelFor(model => model.name)

@Html.EditorFor(model => model.name)

@Html.ValidationMessageFor(model => model.name)

Ezoic



}

@Html.ActionLink("Back to List", "Index")

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

Ezoic

**Solution:**

Here goes the complete example -

Lets create a simple model -

public class Details

{

public string Name { get; set; }

public string Email { get; set; }

}

Let's generate a few Actions utilizing **AJAX BEGINFORM** to perform GET and POST requests.

static List

Next, we can generate a basic interface that will provide a location for the function Ajax.Beginform to be displayed.

Ezoic

@model RamiSamples.Controllers.Details

@{

ViewBag.Title = "Ajax";

}

**Ajax**

@using (Ajax.BeginForm("SaveData", new AjaxOptions()

{

InsertionMode = InsertionMode.Replace,

UpdateTargetId = "dane"

}))

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

Details

@Html.LabelFor(model => model.Name)

@Html.EditorFor(model => model.Name)

@Html.ValidationMessageFor(model => model.Name)

@Html.LabelFor(model => model.Email)

Ezoic

@Html.EditorFor(model => model.Email)

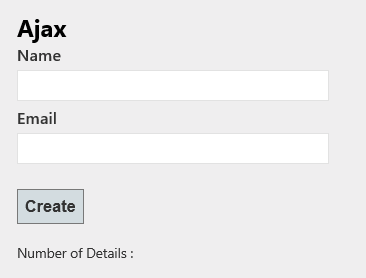
@Html.ValidationMessageFor(model => model.Email)



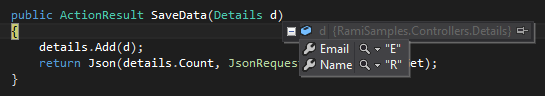
}

Number of Details :

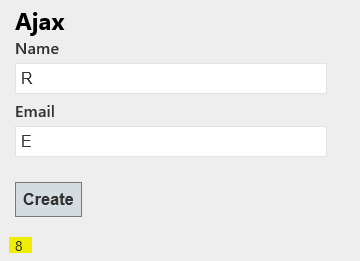
Now when the page gets rendered -



Upon inputting data, and subsequently clicking the "create" button -



Following that, the page will refresh automatically, displaying the quantity of new additions as illustrated.



Use JavaScript to submit Ajax.BeginForm(), Based on the edits to the question and the comments, you wanting to call an ajax function in the OnBegin() option which wont work because ajax is asynchronous. Instead, use jQuery.ajax() to submit you form rather than the Ajax.BeginForm() method (and save yourself the extra overhead of including …

**Tags:**

[submit form using ajax in aspnet mvc](https://copyprogramming.com/t/submit-form-using-ajax-in-aspnet-mvc)[ajax calls with fetch api in net core](https://copyprogramming.com/t/ajax-calls-with-fetch-api-in-net-core)

**Ajax Calls with Fetch API in .NET Core 3.1**

**Question:**

After switching to .NET Core 3.1 , I faced a challenge with updating various components. Unfortunately, I am unable to make my ajax calls function properly.

My action's object parameter does not bind with any data when an AJAX call is made.

This is the manner in which I am placing the call on my Razor page.

fetch('/Administration/UpdateUserStatus', {

method: 'POST',

credentials: 'include',

headers: {

'Content-Type': 'application/json',

'RequestVerificationToken': $('input[name="\_\_RequestVerificationToken"]').val()

},

body: JSON.stringify(data)

})

And my controller action

[HttpPost]

public IActionResult UpdateUserStatus([FromBody]User user) {

...

}

Upon debugging the aforementioned code, it has been observed that the data received from the ajax call does not bind to the designated object, namely **User** , as it appears in a different format ( **null** ) than expected ( **user** ).

After taking out **[FromBody]** , **null** is no longer present, but the information still fails to link. The features are either set as null or the default value.

The .NET Core 2.1 was functioning without any issues.

Do I have to include it in **Startup.cs** or is there a different configuration that I am overlooking?

Thanks

**Solution:**

After encountering a problem, I was able to resolve it by adding a reference to a package called **Microsoft.AspNetCore.Mvc.NewtonsoftJson** (available on NuGet) since Json.NET was removed from .NET Core 3.0. To ensure that the issue was resolved, I added **.AddNewtonsoftJson()** to Razor Pages under the **ConfigureServices** method.

services.AddRazorPages().AddNewtonsoftJson();

Following that, the system is functioning as it did previously.

Docs

How to post ASP.NET MVC Ajax form using JavaScript, Works in Firefox and Safari: # ('form#ajaxForm')trigger ('submit'); Also, if you trigger ('submit') in Chrome or IE, it causes the entire page to be posted rather than doing an AJAX behavior. What works for all browsers is removing the onsubmit event behavior and just calling submit () on the form itself.

**Bootstrap modal+ajax+jquery+validation+.net core 2.0**

**Question:**

My goal is to develop a simple application for learning Bootstrap and MVC, specifically utilizing the AdminLTE template with .NET Core 2.0. At first, everything was functioning properly with MVC alone. However, I wanted to expand my knowledge and began implementing Ajax and jQuery for data transfer during the create and edit process. To accomplish this, I utilized the bootstrap modal plugin. The primary issue I encountered was the inability to display error messages during server-side validation, despite being able to validate on the client-side.

\_Layout.cshtml

@\*\*@

@await Component.InvokeAsync("TopMenubar")

@await Component.InvokeAsync("Sidebar")

@await Component.InvokeAsync("ContentHeader")

@await Component.InvokeAsync("Alert")

@RenderBody()

@await Component.InvokeAsync("Footer")

@RenderSection("scripts", required: false)

@\*\*@

Index.cshtml

@model MovieDb.Models.CategoryType

@{

ViewData["Title"] = "Category Type";

}

*Add New Category*

| **Name** |  |
| --- | --- |

×

**Add New Category**

Top of Form

@Html.AntiForgeryToken()

Name:



Bottom of Form

Add

Close

categoryType.js

$(document).ready(function(){

loadData();

});

function loadData(){

$("#DgCategoryType").DataTable(

{

"filter": true,

"orderMulti": false,

"ajax":

{

"url":"/CategoryTypes/LoadData",

"type":"GET",

"dataType":"JSON"

},

"columns":[

{"data":"Name"},

{ "data":"Id",

"render":function(data){

return "Edit | Delete";

}

}

],

"columnDefs":

[

{

"targets": [1],

"searchable": false

},

{"width":"10%","targets":[1]}

]

}

);

}

function Add(){

/\*var frm=document.getElementById("createCTForm");

var data=toJSONString(frm);\*/

var data=$("#createCTForm").serialize();

$.ajax({

type:"POST",

url:"CategoryTypes/Create",

data:data,

success:function(result){

if(result.success)

{

$("#createNewCategoryType").modal("hide");

}

},

});

}

function clearTextBox(){

$("#Name").val("");

}

Model: CategoryType.cs

public class CategoryType

{

public int Id { get; set; }

[Required(ErrorMessage ="Category name required")]

public string Name { get; set; }

}

CategoryTypesController.cs

public class CategoryTypesController : BaseController

{

private readonly ApplicationDbContext \_context;

public CategoryTypesController(ApplicationDbContext context)

{

\_context = context;

}

protected override void Dispose(bool disposing)

{

if (disposing)

\_context.Dispose();

base.Dispose(disposing);

}

// GET: CategoryTypes

public IActionResult Index()

{

AddPageHeader("Category Types");

return View(new CategoryType());

}

//GET JSON data for loading the datatable

public async Task LoadData()

{

var data = await \_context.CategoryTypes.ToListAsync();

return Json(new { data } );

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task Create([Bind("Id,Name")] CategoryType categoryType)

{

if (ModelState.IsValid)

{

\_context.Add(CategoryType);

await \_context.SaveChangesAsync();

return Json(new { success=true });

}

return Json(new { success=false });

}

}

Upon executing the aforementioned code, the new category will be included in the database. Nevertheless, if the name is not entered in the modal input box, an error will be generated by the controller but the validation message will not be displayed.

what I am doing wrong?

**Solution 1:**

Having encountered a similar issue, I adopted a proactive approach and developed a small validation tool that I refer to as a mini-library. You can access the code for this tool on JsFiddle, where you will also find a functional demonstration.

At present, the code includes validations for mandatory fields, regular expressions, and range, equals . If you require additional checks, feel free to add them yourself or let me know in the comments, and I'll do my best to assist you.

How it works:

Generate a JavaScript file within your project and copy over the minimum required code from the CustomValidationScript region provided in the fiddle link. Afterwards, make sure to incorporate this file into your \_layout.cshtml.

2.Setup some rules:

var validationRules = [{

ruleName: "elementId1ShoudBeRequired", // make sure this

name will be unique within your rules

ruleForElementId: "#elementId1",

check: "required",

comparerValue: true,

message: "This input is required! Please make sure to input a value!"

},

{

ruleName: "elementId3ShouldBePhoneNr",

ruleForElementId: "#elementId2",

check: "regex",

comparerValue: /^(\+\d{1,2}\s)?\(?\d{3}\)?[\s.-]\d{3}[\s.-]\d{4}$/,

message: "Please provide a valid phone number!"

},

{

ruleName: "elementId4\_ShouldBeEqualWith\_5",

ruleForElementId: "#elementId2",

check: "equals",

comparerValue: 5,

message: "This input should be 5!"

}

];

var rangeRules = [{

ruleName: "elementId2ShoudBeInRange",

ruleForElementId: "#elementId2",

check: "range",

comparerValue: [1, 8],

message: "Please provide a value between 1 and 8 !"

}];

1. In your js file / script initialize the validator outside of $(document).ready() function:
2. var validator = new CustomValidation(validationRules);
3. // you can pass the validation rules in the initialization, or send a empty array if you wish to add the rules later
4. // var validator = new CustomValidation([]);

1. *(optional step)*Add or remove more validation rules:
2. // Add rules
3. // you can add them by passing an array of rules
4. validator.addcustomValidationRules(rangeRules);
5. // or by passing one rule
6. validator.addcustomValidationRule(
7. {
8. ruleName: addRule.ruleName,
9. ruleForElementId: addRule.ruleForElementId,
10. check: addRule.check,
11. comparerValue: addRule.comparerValue,
12. message: addRule.message
13. }
14. );
15. // if the rule(s) already exist(s), an error will be thrown
16. // Remove rules
17. // you can also remove some rules by ruleName, or sending an array of rules
18. // for example we can remove rangeRules entirely like :
19. validator.removeCustomValidationRules(rangeRules);
20. // or we can remove individual rules by name like so:
21. validator.removeCustomValidationRule("elementId2ShoudBeInRange");

1. -

Validate the inputs:

-

**var result = validator.validate(validator); // will return an object like this:{ formIsvalid : true / false, validationResults: [ {elementId: elementId,message: message}] }**

-

HTML page example:

-

<div class="container">

<div class="form-group">

<label>Input for elementId1 </label>

<input id="elementId1" class="validate" />

<span class="error\_span col-md-12"></span>

</div>

<div class="form-group">

<label>Input for elementId2 </label>

<input id="elementId2" class="validate" />

<span class="error\_span"></span>

</div>

<button class="btn btn-outline-success" id="save">Save</button>

</div>

<script type="text/javascript">

// you can validate the input inside listeners or inside whatever pieces of code

// example onClick

$("#save").on("click", function() {

// the validation happerns here

var result = validator.validate(validator);

if (result.formIsValid) {

// your code ...

// can be ajax call to send the form to the server

}

else {

// clear error\_spans that might have been fixed between clicks

updateUi(result.validationResults, "form-group", "error\_span");

}

});

// example onKeyUp

$(".validate").on("keyup", function() {

var result = validator.validate(validator);

updateUi(result.validationResults, "form-group", "error\_span");

});

</script>

The custom function called updateUi, which can also be located in JsFiddle, is responsible for showing or hiding the error spans by updating them.

That's it!

It successfully passed my tests with ajax calls, select2, and bootstrap 4.

If you come across any areas that can be enhanced, please feel free to leave a comment. As I'm not an expert in JavaScript, there could be some errors that I'm unaware of.

Happy coding!

**Solution 2:**

Uncertainty surrounds the reason why the validation didn't work in the aforementioned method. jQuery- unobtrusive-ajax - was utilized to address the issue, and additional information can be found at https://dotnetthoughts.net/ jquery-unobtrusive-ajax-helpers -in-aspnet-core/.

Here is a tutorial on how to utilize Ajax.Beginform for validating a pop-up modal in C# MVC: https://qawithexperts.com/article/asp.net/validate-pop-up-modal-using-ajaxbeginform-in-c-mvc/52

**Solution 3:**

To resolve the issue, ensure that both the jquery-validate and jquery-validation-unobtrusive scripts are connected to the view accessed via the modal.

You require it as partial views do not have access to scripts or script tags on the layout view.

Javascript - bootstrap, You can fix it by linking the jquery-validate and the jquery-validation-unobtrusive scripts with the view that you're calling through the modal. You need that because any view acting as a partial view does not have access to scripts or script tags on the layout view. Share. edited Jun 29, 2021 at 7:28.

**Read other technology post:** [Retrofit: passing multiple GET parameter](https://copyprogramming.com/howto/retrofit-passing-multiple-get-parameter)

**Related posts:**

[An Illustration of MVC JavaScript Code for Submitting Ajax Forms](https://copyprogramming.com/howto/submit-ajax-form-javascript-mvc-code-example)

[Resetting an HTML form using the <form> tag](https://copyprogramming.com/howto/html-html-begin-form-reset-form)

[Utilizing Ajax to Post Partial Views in ASP.NET MVC](https://copyprogramming.com/howto/asp-net-mvc-partial-view-ajax-post)

[Unsuccessful Unobtrusive Validation for Dynamically Added Partial View in Ajax.BeginForm](https://copyprogramming.com/howto/unobtrusive-validation-on-dynamically-added-partial-view-in-ajax-beginform-not-working)

[Troubleshooting: Inability of Ajax.BeginForm to Transmit Form Values](https://copyprogramming.com/howto/why-ajax-beginform-does-not-pass-form-values)

[Utilizing Razor Views and PartialViews with Ajax Postbacks in ASP.net MVC3](https://copyprogramming.com/howto/asp-net-mvc3-razor-views-and-partialviews-with-ajax-postbacks)

[Utilizing Ajax Beginform in Asp.net MVC 4: A Guide](https://copyprogramming.com/howto/how-to-use-simple-ajax-beginform-in-asp-net-mvc-4)

[Is it possible to implement JQuery in the layout and view of MVC?](https://copyprogramming.com/howto/can-a-jquery-be-used-in-a-layout-and-view-in-mvc)

[Posting entire form using Ajax in MVC 5](https://copyprogramming.com/howto/mvc-5-ajax-posting-whole-form)

[Form submission with file upload using Ajax.BeginForm in MVC5](https://copyprogramming.com/howto/mvc5-ajax-beginform-upload-form-with-files)

[Using jQuery Ajax Unobtrusive to Send asp-route-data to a Page Handler](https://copyprogramming.com/howto/passing-asp-route-data-to-a-page-handler-using-jquery-ajax-unobtrusive)

[Returning a JSON response to an Ajax.BeginForm: What are the options?](https://copyprogramming.com/howto/how-can-i-return-a-json-result-to-a-ajax-beginform)

[Troubleshooting ASP.NET Core and JQuery's Unobtrusive AJAX That Isn't Functional](https://copyprogramming.com/howto/asp-net-core-and-jquery-unobtrusive-ajax-not-working)

[Ezoic](https://copyprogramming.com/howto/asp-net-core-and-jquery-unobtrusive-ajax-not-working)

[Streamline Your Workflow: Automate ...](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Play](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Unmute](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Loaded: 4.18%](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Remaining Time -5:35](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Fullscreen](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Advertisement: 2:02](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[Play Video](https://copyprogramming.com/howto/call-javascript-from-within-an-asp-net-mvc-partial-view)

[click me](https://letmeflutter.com/category/python-programming/)

[Streamline Your Workflow: Automate the Boring Stuff with Python](https://humix.com/redirect?url=https%3A%2F%2Fletmeflutter.com%2Fhumix%2Fvideo%2F1f04a72068fdb0b6bfe27e85f737e6173ca236ea6ce3b1cffe2a68cb3961b061)

Invoke JavaScript within a Partial View of ASP.NET MVC

[Submitting a form in Bootstrap modal using MVC could be the rephrased](https://copyprogramming.com/howto/bootstrap-modal-form-submit-in-mvc)

[Detecting the Fully Opened Sidebar in AdminLTE](https://copyprogramming.com/howto/adminlte-know-when-sidebar-fully-open)

[An Example of Code Designing for MVC Framework could be the rephrased](https://copyprogramming.com/howto/how-to-design-an-mvc-framework-code-example)

[Utilizing HTML Code Across Multiple Pages](https://copyprogramming.com/howto/how-to-reuse-the-html-code-in-every-html-page)

[Using Partial View with AJAX in ASP.NET Core MVC Using C#](https://copyprogramming.com/howto/csharp-asp-net-core-mvc-ajax-partial-view)

[Example Code for a Modal Form in Bootstrap Using HTML and Step-by-Step Instructions](https://copyprogramming.com/howto/html-step-bootstrap-form-modal-form-code-example)

[Implementing Jquery on a Webpage: Tips and Tricks](https://copyprogramming.com/howto/how-can-i-get-jquery-working-for-this-page)

[Complete postback performed by Ajax.BeginForm](https://copyprogramming.com/howto/ajax-beginform-does-a-full-postback)

[Using Javascript AJAX to Search for Twitch Streams via API](https://copyprogramming.com/howto/twitch-api-search-stream-via-javascript-ajax)

[Loading a model with two distinct object lists within a modal](https://copyprogramming.com/howto/trying-to-load-a-model-that-contains-2-lists-of-different-objects-in-a-modal)

[Using jQuery to Close the Parent Modal Window](https://copyprogramming.com/howto/close-button-parent-modal-using-jquery)

[Modal dialog in MVC causing dysfunction in client form validation](https://copyprogramming.com/howto/client-form-validation-not-working-with-modal-dialog-in-mvc)

[Chrome compatibility issue with Ajax.BeginForm in MVC 5](https://copyprogramming.com/howto/ajax-beginform-not-working-in-chrome-mvc-5)

[Preventing Form Submission Duplication in ASP.NET MVC](https://copyprogramming.com/howto/how-to-avoid-multiple-submit-of-form-in-asp-net-mvc)

[ASP.Net Core: The Functionality of Unobtrusive Validation is Inoperative](https://copyprogramming.com/howto/asp-net-core-unobtrusive-validation-not-working)

[ASP.NET MVC Validation](https://copyprogramming.com/howto/asp-net-mvc-validation)