

Returning Partialview & JSON from MVC 5 Controller

Asked 3 years, 8 months ago Active 3 years, 8 months ago Viewed 5k times



2



1



In an MVC5 project, I open a modal dialog and in case there is an exception, I want to open this dialog and display a message on a div in this dialog. As far as I see, I should follow the approach as rendering partialview into a string, but most of the examples does not work in MVC5 as on [Return Partial View and JSON from ASP.NET MVC Action](#). Is there any similar or better approach working for MVC5?

json

ajax

asp.net-mvc

asp.net-mvc-4

asp.net-mvc-partialview

edited May 23 '17 at 12:33



Community ♦

1 ● 1

asked Sep 21 '16 at 19:51



Jack

4,122 ● 14 ● 63 ● 131

1 you want to show the errors on the modal and not the partial view it self with errors right? – Monah Sep 21 '16 at 20:06

Actually I render the partial view for create, update, etc. and if there is an error in the Create action method in the controller, I want to render `_Error` partialview instead of `_Create` and display some messages (returned from the action method as json) in the `_Error` partialview. – Jack Sep 21 '16 at 20:10

2 Answers

Active

Oldest

Votes



4



You can do the following

Solution 1 (using partial views)

```
[HttpPost]
public ActionResult YourAction(YourModel model)
{
    if(model!=null && ModelState.IsValid)
    {
        // do your staff here
        Response.StatusCode = 200; // OK
        return PartialView("ActionCompleted");
    }
}
```

```

    }
    else
    {
        Response.StatusCode = 400; // bad request
        // ModelState.ToErrors() : is an extension method that convert
        // the model state errors to dictionary
        return PartialView("_Error", ModelState.ToErrors());
    }
}

```

Your partial view should look like:

```

<div id="detailId">
    <!-- Your partial details goes here -->
    ....
    <button type="submit" form="" value="Submit">Submit</button>

</div>

```

your script

```

<script>
    $(document).ready(function(){
        $('#formId').off('submit').on('submit', function(e){
            e.preventDefault();
            e.stopPropagation();

            var form = $('#formId');
            $.ajax({
                url: form.attr('action'),
                data: form.serialize(),
                method: 'post',
                success : function(result){
                    $('#detailId').replaceWith(result);
                    // another option you can close the modal and refresh your data.
                },
                error: function(data, status, err){
                    if(data.status == 400){
                        $('#detailId').replaceWith(data.responseText);
                    }
                }
            });
        });
    });
</script>

```

Solution 2 (using Json)

in your action

```
[HttpPost]
public ActionResult YourAction(YourModel model)
{
    if(model!=null && ModelState.IsValid)
    {
        // do your staff here
        return Json(new {status = 200,
            //...any data goes here... for example url to redirect
            url=Url.Content("YourRedirectAction","Controller")},
        );
    }
    else
    {
        return Json( new {status= 400,errors = ModelState.ToErrors()});
    }
}
```

and your script should look like

```
<script>
$(document).ready(function(){
    $('#formId').off('submit').on('submit', function(e){
        e.preventDefault();
        e.stopPropagation();

        var form = $('#formId');
        $.ajax({
            url: form.attr('action'),
            data: form.serialize(),
            method: 'post',
            success : function(result){
                if(result.status==200) { // OK
                    // you might load another action or to redirect
                    // this conditions can be passed by the Json object
                }
                else{ // 400 bad request
                    // you can use the following toastr based on your comment
                    // http://codeseven.github.io/toastr/demo.html
                    var ul = $('<ul>')
                    for(var error in result.errors)
                    {

```

```

        ul.append('<li><b>' + error.Key + '</b>:' + error.Value
    + '</li>');
    }
    toastr["warning"](ul[0].outerHTML);
}
});
});
});
</script>

```

Finally, if you want the extension `ModelState.ToErrors()`

```

public static IEnumerable ToErrors(this ModelStateDictionary modelState)
{
    if (!ModelState.IsValid)
    {
        return ModelState.ToDictionary(kvp => kvp.Key,
            kvp => kvp.Value.Errors
                .Select(e => e.ErrorMessage).First()
                .Where(m => m.Value.Count() > 0));
    }
    return null;
}

```

hope this will help you

edited Sep 22 '16 at 14:09

answered Sep 21 '16 at 20:37



Monah

5,973 ● 5 ● 17 ● 49

Thanks a lot, I will give a try. Voted+ – **Jack** Sep 21 '16 at 20:39

Not actually, but I preferred to use another approach and display toast message without opening modal dialog. Thanks... – **Jack** Sep 22 '16 at 13:40

toast message will contains the errors only, I asked you on your question comments how you want it, i will update the answer – **Monah** Sep 22 '16 at 13:48

@binary I updated the answer to include another approach – **Monah** Sep 22 '16 at 14:09

This is a valid example, i have used this technique many times. If it is a simple get call than i would suggest make a partial view of

0

data you want to show and call it via jquery with code below.

```
$( "#result" ).load("@Url.Action("Account","HelloPartial")");
```

this will load the partial view its self in the popup. you won't have to convert it to strings.



answered Sep 21 '16 at 20:24



[Zain Ahmad Khan](#)

309 ● 2 ● 20