**Updated Notification Script (Index.cshtml)**

html

Copy

@section Scripts {

<script>

// Initialize notification container first

var notification = $("#notification").kendoNotification({

position: {

pinned: true,

top: 50,

right: 30

},

autoHideAfter: 4000,

stacking: "down"

}).data("kendoNotification");

// Simplified error handler

function onDataSourceError(e) {

var errorMessage = "Operation failed: ";

if (e.errors && e.errors.message) {

errorMessage += e.errors.message;

} else if (e.xhr && e.xhr.responseText) {

try {

var errorData = JSON.parse(e.xhr.responseText);

errorMessage += errorData.title || errorData.message || "Unknown error";

} catch {

errorMessage += "Server error";

}

}

notification.error(errorMessage);

}

// Unified request handler

function onRequestEnd(e) {

if (e.type === "create" && e.response) {

notification.success("Record created successfully");

} else if (e.type === "update" && e.response) {

notification.success("Record updated successfully");

} else if (e.type === "destroy" && e.response) {

notification.success("Record deleted successfully");

}

}

</script>

<!-- Keep your existing autocomplete script -->

<script>

function getAutoCompleteParams() {

var input = $("input[data-role='autocomplete']:focus");

return {

columnName: input.attr("name"),

searchText: input.val() || ""

};

}

</script>

}

Run HTML

**Key Fixes:**

1. **Simplified Notification Initialization**:

javascript

Copy

var notification = $("#notification").kendoNotification({

position: {

pinned: true,

top: 50,

right: 30

},

autoHideAfter: 4000,

stacking: "down"

}).data("kendoNotification");

1. **Removed Custom Templates** (using default styling instead):
   * Removed the complex template definitions
   * Using built-in success() and error() methods
2. **Robust Error Handling**:

javascript

Copy

function onDataSourceError(e) {

var errorMessage = "Operation failed: ";

if (e.errors && e.errors.message) {

errorMessage += e.errors.message;

} else if (e.xhr && e.xhr.responseText) {

try {

var errorData = JSON.parse(e.xhr.responseText);

errorMessage += errorData.title || errorData.message || "Unknown error";

} catch {

errorMessage += "Server error";

}

}

notification.error(errorMessage);

}

1. **Keep Grid DataSource Configuration**:

javascript

Copy

.DataSource(dataSource => dataSource

.Ajax()

.Events(events => events

.Error("onDataSourceError")

.RequestEnd("onRequestEnd")

)

// Rest of your existing configuration

)

**Why This Works:**

1. **Avoids Template Syntax Issues** by using Kendo's built-in methods
2. **Handles ASP.NET Core Error Responses** properly
3. **Works with Your Existing Controller Logic** without modifications
4. **Maintains Security** by not exposing raw error details

**Additional Recommendations:**

1. **Update Kendo Packages** to latest version if possible:

bash

Copy

Update-Package Telerik.UI.for.AspNet.Core

1. **Add Server-Side Validation** to your controller:

csharp

Copy

[HttpPost]

[AllowAnonymous]

public async Task<IActionResult> CreateNscPartsUsed(

[DataSourceRequest] DataSourceRequest request,

[FromForm] NscPartsUsedViewModel nscPartsUsed)

{

if (!ModelState.IsValid)

{

return BadRequest(new {

message = "Invalid data submitted",

errors = ModelState.Values

.SelectMany(v => v.Errors)

.Select(e => e.ErrorMessage)

});

}

// Rest of your existing code

}

This solution maintains all your existing functionality while fixing the template error and providing better error handling.