To ensure notifications display properly, let's refine the implementation. The key issue is that while we're using notification.success()/error(), we need to ensure proper event handling and response parsing. Here's the corrected solution:

**1. Update Notification Initialization (Index.cshtml)**

html

Copy

<div id="notification"></div>

@section Scripts {

<script>

// Initialize notification FIRST

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

position: {

pinned: true,

top: 50,

right: 30

},

autoHideAfter: 4000,

stacking: "down"

}).data("kendoNotification");

// Error handler

function onDataSourceError(e) {

let errorMessage = "Operation failed: ";

if (e.errorThrown) {

errorMessage += e.errorThrown;

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

errorMessage += e.xhr.responseText;

}

notification.error(errorMessage);

}

// Success handler

function onRequestEnd(e) {

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

notification.success("Record created successfully");

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

notification.success("Record updated successfully");

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

notification.success("Record deleted successfully");

}

}

</script>

}

Run HTML

**2. Update Controller Responses (NSCPartsController.cs)**

Modify your return statements to include JSON objects:

csharp

Copy

// In CreateNscPartsUsed

return Json(new { success = true });

// In UpdateNscPartsUsed

return Json(new { success = true });

// In DeleteNscPartsUsed

return Json(new { success = true });

// For errors

return StatusCode(500, new { error = "Your error message" });

**3. Verify Grid DataSource Configuration**

Ensure your grid has these event bindings:

csharp

Copy

.DataSource(dataSource => dataSource

.Ajax()

.Events(events => events

.Error("onDataSourceError")

.RequestEnd("onRequestEnd")

)

// rest of your config

)

**Key Fixes:**

1. **Explicit Success Messages**: Now shows notifications based on operation type regardless of response content
2. **Simplified Error Handling**: Directly uses error information from the DataSource event
3. **Structured Responses**: Controller returns JSON objects that Kendo can parse