**1. Updated Index.cshtml (Key Changes):**

html

Copy

<script id="template" type="text/x-kendo-template">

<!-- Add unique identifier to template container -->

<div id="detail-template-#=RebuiltStockNum#"></div>

</script>

<script>

// Global variable to track current expanded row

var currentExpandedRow = null;

// Unified function to check unsaved changes

function hasUnsavedChanges(rebuiltStockNum) {

let hasChanges = false;

const tabStrip = $("#tabstrip-container-" + rebuiltStockNum).data("kendoTabStrip");

if (tabStrip) {

const contentElement = tabStrip.contentElement(tabStrip.select());

$(contentElement).find(".k-grid").each(function() {

const grid = $(this).data("kendoGrid");

if (grid && grid.dataSource.hasChanges()) {

hasChanges = true;

return false; // Break loop

}

});

}

return hasChanges;

}

grid.bind("detailInit", function(e) {

const rebuiltStockNum = e.data.RebuiltStockNum;

console.log("Detail init for:", rebuiltStockNum);

// Generate unique IDs for all elements

const container = e.detailCell.find(`#detail-template-${rebuiltStockNum}`);

container.html(`

<div id="tabstrip-container-${rebuiltStockNum}">

<ul>

<li class="k-state-active">Labour Details</li>

<!-- Other tabs... -->

</ul>

<div id="labour-details-content-${rebuiltStockNum}"></div>

<!-- Other content divs with unique IDs... -->

</div>

`);

// Initialize TabStrip with unique selector

const tabStrip = $(`#tabstrip-container-${rebuiltStockNum}`).kendoTabStrip({

// ... existing config ...

select: function(e) {

if (hasUnsavedChanges(rebuiltStockNum)) {

e.preventDefault();

$("#notification").data("kendoNotification").show("Save changes before switching tabs!", "warning");

return;

}

// ... rest of select handler ...

}

}).data("kendoTabStrip");

// ... rest of detailInit ...

});

// Handle row expansion/collapse

grid.bind("detailExpand", function(e) {

if (currentExpandedRow && currentExpandedRow !== e.masterRow) {

if (hasUnsavedChanges(currentExpandedRow.data("RebuiltStockNum"))) {

e.preventDefault();

$("#notification").data("kendoNotification").show("Save changes before switching rows!", "warning");

return;

}

grid.collapseRow(currentExpandedRow);

}

currentExpandedRow = e.masterRow;

currentExpandedRow.data("RebuiltStockNum", e.data.RebuiltStockNum);

});

grid.bind("detailCollapse", function(e) {

if (hasUnsavedChanges(e.data.RebuiltStockNum)) {

e.preventDefault();

$("#notification").data("kendoNotification").show("Save changes before closing!", "warning");

}

});

</script>

Run HTML

**2. Updated \_labourDetailsRebuiltParts.cshtml:**

html

Copy

@(Html.Kendo().Grid<LabourDetailsRebuiltPartsViewModel>()

.Name("LabourRebuiltPartGrid\_#=RebuiltStockNum#") <!-- Unique grid name -->

.Columns(columns => {/\* ... unchanged ... \*/})

.DataSource(dataSource => dataSource

.Ajax()

.Model(model => model.Id(c => c.Id))

.Read(read => read.Url(/\* ... \*/ + "#=RebuiltStockNum#"))

// Other operations with rebuiltStockNum parameter

)

)

Run HTML

**Key Implementations:**

1. **Unique ID System:**
   * All dynamic elements now include RebuiltStockNum in their IDs
   * Prevents DOM conflicts between multiple expanded rows
   * Affects: template container, tabstrip container, content divs, and child grids
2. **Row Management:**
   * Track current expanded row with currentExpandedRow variable
   * Auto-collapse previous row using detailExpand event
   * Store RebuiltStockNum in row data attribute for easy retrieval
3. **Unsaved Changes Handling:**
   * hasUnsavedChanges() checks all grids in current tabstrip
   * Uses Kendo's dataSource.hasChanges()
   * Prevents tab switching/row collapse with unsaved changes
   * Shows warning notifications using Kendo notification widget
4. **Grid Naming:**
   * Child grids now have unique names using #=RebuiltStockNum#
   * Prevents JS conflicts when multiple instances exist

**Notification System Enhancements:**

javascript

Copy

// Initialize notification once

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

position: { pinned: true, top: 20, right: 20 },

autoHideAfter: 5000,

stacking: "down"

}).data("kendoNotification");

// Usage examples

notification.show("Saved successfully!", "success");

notification.show("Unsaved changes detected!", "warning");

notification.show("Operation canceled", "info");

**Critical Security Note:**  
Always validate RebuiltStockNum parameter server-side in your controller actions to prevent parameter tampering:

csharp

Copy

public IActionResult LabourDetailsPartial(string rebuiltPartNum)

{

if (!IsValidStockNumber(rebuiltPartNum))

{

return BadRequest("Invalid stock number");

}

// ... rest of action ...

}