To implement the requested changes, we need to modify the RebuiltParts/Index.cshtml file to include a Telerik TabStrip with two tabs. The first tab will contain the existing content of RebuiltParts/Index.cshtml, and the second tab will load the ArchivedIndex.cshtml view.

Here are the updated files with comments and exception handling:

**Updated RebuiltParts/Index.cshtml**

csharp

Copy

@using BCES.Models.Parts

@using BCES.Models.Common

@model IEnumerable<BCES.Models.Parts.RebuiltPartsViewModel>

@{

ViewData["Title"] = "Rebuilt Parts";

}

<h4>Rebuilt Parts</h4>

@(Html.Kendo().TabStrip()

.Name("rebuildPartsTabStrip")

.Animation(animation => animation.Open(effect => effect.Fade(FadeDirection.In)))

.Items(tabstrip =>

{

tabstrip.Add().Text("Rebuilt Parts")

.Selected(true)

.Content(@<text>

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

.Name("RebuiltPartsGrid")

.Columns(columns =>

{

columns.Bound(c => c.RebuiltStockNum).Title("Rebuilt Stock Code").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.MmsStockCode).Title("MMS Stock Code").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.DetailedDesc).Title("Description").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.Keyword).Title("Keyword").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.JobNumber).Title("Job Number").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.CorePartNum).Title("Core Code").Filterable(ftb => ftb.Cell(cell => cell.Operator("contains").SuggestionOperator(FilterType.Contains)));

columns.Bound(c => c.CoreCharge).Title("Core Cost").Filterable(ftb => ftb.Cell(cell => cell.Operator("gte"))).Format("{0:C2}").EditorTemplateName("Currency");

columns.Bound(c => c.BuyNewCost).Title("Buy Cost").Filterable(ftb => ftb.Cell(cell => cell.Operator("gte"))).Format("{0:C2}").EditorTemplateName("Currency");

columns.Bound(c => c.RemanCost).Title("Remanufactured Cost").Filterable(ftb => ftb.Cell(cell => cell.Operator("gte"))).Format("{0:C2}").EditorTemplateName("Currency");

columns.Command(command =>

{

command.Edit();

command.Custom("Archive").Click("onArchiveClick"); // Add a custom "Archive" button

command.Custom("Save to Excel").Click("onSaveToExcelClick");

command.Custom("Email Estimate").Click("onEmailEstimateClick");

}).Title("Actions");

})

.ToolBar(toolbar => { toolbar.Create(); })

.Editable(editable => editable.Mode(GridEditMode.InLine))

.Pageable()

.Sortable()

.ClientDetailTemplateId("template")

.Resizable(r => r.Columns(true))

.Filterable(ftb => ftb.Mode(GridFilterMode.Row))

.DataSource(dataSource => dataSource

.Ajax()

.Model(model =>

{

model.Id(c => c.RbMasterlistId);

model.Field(c => c.RbMasterlistId);

model.Field(c => c.RebuiltStockNum).Editable(true);

model.Field(c => c.MmsStockCode).Editable(true);

model.Field(c => c.VehicleSeries).DefaultValue(new List<BCES.Models.Parts.BusesModel>()).Editable(true);

model.Field(c => c.LabourDetailsRebuiltParts).DefaultValue(new List<BCES.Models.Parts.LabourDetailsRebuiltPartsViewModel>()).Editable(false);

})

.Read(read => read.Url(Url.Action("GetRebuiltPartsView", "RebuiltParts")).Type(HttpVerbs.Get))

.Create(create => create.Url(Url.Action("AddRebuiltPartsView", "RebuiltParts")).Type(HttpVerbs.Post))

.Update(update => update.Url(Url.Action("UpdateRebuiltPartsView", "RebuiltParts")).Type(HttpVerbs.Post))

)

)

@Html.Partial("\_EmailEstimatePopup")

<div id="estimates-container"></div>

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

</text>);

tabstrip.Add().Text("Archived Parts")

.Content(@<text>

@try

{

@Html.Partial("ArchivedIndex")

}

catch (Exception ex)

{

<div class="alert alert-danger">Error loading archived parts: @ex.Message</div>

}

</text>);

})

)

@section Scripts {

<script>

$(document).ready(function () {

var grid = $("#RebuiltPartsGrid").data("kendoGrid");

// Initialize Kendo Notification for displaying messages

var notificationElement = $("#notification");

if (notificationElement.length === 0) {

console.error("Notification element not found in the DOM.");

} else {

var notification = notificationElement.kendoNotification({

position: {

pinned: true,

top: 50,

right: 50

},

autoHideAfter: 5000, // Hide after 5 seconds

stacking: "down"

}).data("kendoNotification");

// Bind the Save event for the grid

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

// This event is triggered when the user clicks the Update button

$.post('@Url.Action("UpdateRebuiltPartsView", "RebuiltParts")', { rebuiltPartNum: e.model.RebuiltStockNum })

.done(function (data) {

if (notification) {

notification.show("Changes saved successfully!", "success");

} else {

console.error("Notification is not initialized.");

}

})

.fail(function (xhr, status, error) {

console.error("Error saving changes: ", error);

if (notification) {

notification.show("An error occurred while saving changes. Please try again.", "error");

} else {

console.error("Notification is not initialized.");

}

});

});

// Bind the Cancel event for the grid

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

// This event is triggered when the user clicks the Cancel button

$.post('@Url.Action("CancelDetails", "RebuiltParts")', { rebuiltPartNum: e.model.RebuiltStockNum })

.done(function (data) {

if (notification) {

notification.show("Changes discarded!", "info");

} else {

console.error("Notification is not initialized.");

}

})

.fail(function (xhr, status, error) {

console.error("Error discarding changes: ", error);

if (notification) {

notification.show("An error occurred while discarding changes. Please try again.", "error");

} else {

console.error("Notification is not initialized.");

}

});

});

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

console.log("Detail initialization started for RebuiltStockNum: " + e.data.RebuiltStockNum);

var tabUrlLabourDetail = '@Url.Action("LabourDetailsPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabUrlLabourHourSummary = '@Url.Action("LabourHourSummaryPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabUrlMaterialCostSummary = '@Url.Action("MaterialCostSummaryPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabScPartDetail = '@Url.Action("ScPartsUsedInRbParts", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabUrlNscDetail = '@Url.Action("NscDetailsPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabUrlRbPartDetail = '@Url.Action("RbDetailsPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

var tabUrlTotalCost = '@Url.Action("TotalCostPartial", "RebuiltParts")' + '?rebuiltPartNum=' + e.data.RebuiltStockNum;

// Get the container for the detail template

var container = e.detailCell.find("#detail-template-" + e.data.RebuiltStockNum);

// Load the TabStrip structure

container.html(`

<div id="tabstrip-container-${e.data.RebuiltStockNum}">

<ul>

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

<li>Labour Hour</li>

<li>Material: Cost</li>

<li>Material: SC Parts</li>

<li>Material: NSC Parts</li>

<li>Material: RB Parts</li>

<li>Total Cost</li>

</ul>

<div id="labour-details-content"></div>

<div id="labour-hour-summary-content"></div>

<div id="material-cost-summary-content"></div>

<div id="material-details-content"></div>

<div id="nsc-details-content"></div>

<div id="rb-part-details-content"></div>

<div id="total-cost-content"></div>

</div>

`);

// Initialize the Kendo TabStrip

var tabStrip = $("#tabstrip-container-" + e.data.RebuiltStockNum).kendoTabStrip({

animation: { open: { effects: "fadeIn" } },

select: function (e) {

var tabIndex = $(e.item).index(); // Get the index of the selected tab

// Check for unsaved changes in the current tab

var currentTabContent = tabStrip.contentElement(tabStrip.select().index());

var currentGrid = $(currentTabContent).find(".k-grid").data("kendoGrid");

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

// Show a warning notification

if (notification) {

notification.show("You have unsaved changes. Click on Update or Cancel changes before clicking next tab.", "warning");

} else {

console.error("Notification is not initialized.");

}

e.preventDefault(); // Prevent the tab switch

return;

}

// Load the partial view content for the selected tab

var tabUrl = tabIndex == 0 ? tabUrlLabourDetail :

tabIndex == 1 ? tabUrlLabourHourSummary :

tabIndex == 2 ? tabUrlMaterialCostSummary :

tabIndex == 3 ? tabScPartDetail :

tabIndex == 4 ? tabUrlNscDetail :

tabIndex == 5 ? tabUrlRbPartDetail :

tabUrlTotalCost;

$.get(tabUrl)

.done(function (data) {

$(e.contentElement).html(data);

kendo.init($(e.contentElement)); // Reinitialize Kendo widgets

// Reset the hasChanges() state for read-only tabs

if (tabIndex === 6) { // Assuming the last tab is the read-only tab

var totalCostGrid = $(e.contentElement).find(".k-grid").data("kendoGrid");

notification.show("tabIndex === 6");

if (totalCostGrid) {

notification.show("inside totalCostGrid");

totalCostGrid.dataSource.read(); // Refresh the data source

}

}

})

.fail(function (xhr, status, error) {

console.error("Error loading tab content: ", error);

if (notification) {

notification.show("An error occurred while loading the tab content. Please try again.", "error");

} else {

console.error("Notification is not initialized.");

}

});

}

}).data("kendoTabStrip");

// Load the first tab's content

$.get(tabUrlLabourDetail)

.done(function (data) {

$("#labour-details-content").html(data);

kendo.init($("#labour-details-content")); // Reinitialize Kendo widgets

})

.fail(function (xhr, status, error) {

console.error("Error loading initial tab content: ", error);

if (notification) {

notification.show("An error occurred while loading the initial tab content. Please try again.", "error");

} else {

console.error("Notification is not initialized.");

}

});

});

}

// Handle form submission

$("#emailEstimateForm").on("submit", function (e) {

e.preventDefault();

var form = $(this);

var url = form.attr("action");

var rebuiltPartNum = $("#subject").val().match(/for (.\*) - Date/)[1]; // Extract rebuiltPartNum from subject

$.ajax({

type: "POST",

url: url,

data: {

to: $("#to").val(),

subject: $("#subject").val(),

body: $("#body").val(),

rebuiltPartNum: rebuiltPartNum

},

success: function (data) {

if (data.success) {

var notification = $("#notification").data("kendoNotification");

notification.show("Email Sent", "success");

// Close the Kendo Window

var window = $("#emailEstimateWindow").data("kendoWindow");

window.close();

} else {

var notification = $("#notification").data("kendoNotification");

notification.show(data.message, "error");

}

},

error: function (xhr, status, error) {

var notification = $("#notification").data("kendoNotification");

notification.show("An error occurred while sending the email.", "error");

}

});

});

});

</script>

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

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

</script>

<script>

function dataBound() {

this.expandRow(this.tbody.find("tr.k-master-row").first());

}

</script>

<script>

function onArchiveClick(e) {

e.preventDefault(); // Prevent the default action

// Get the data item for the clicked row

var dataItem = this.dataItem($(e.currentTarget).closest("tr"));

// Call the controller action with the entire dataItem as JSON

$.ajax({

url: '@Url.Action("ArchiveDetails", "RebuiltParts")',

type: 'POST',

contentType: 'application/json',

data: JSON.stringify(dataItem),

success: function (data) {

// Display a Kendo Notification

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

position: {

pinned: true,

top: 50,

right: 50

},

autoHideAfter: 5000, // Hide after 5 seconds

stacking: "down"

}).data("kendoNotification");

notification.show("Archive is clicked.", "info");

},

error: function (xhr, status, error) {

console.error("Error archiving: ", error);

}

});

}

</script>

<script>

function onSaveToExcelClick(e) {

e.preventDefault();

var dataItem = this.dataItem($(e.currentTarget).closest("tr"));

var rebuiltPartNum = dataItem.RebuiltStockNum;

// Directly trigger the file download

window.location.href = '@Url.Action("SaveToExcel", "Export")' + '?rebuiltPartNum=' + rebuiltPartNum;

}

</script>

<script>

function onEmailEstimateClick(e) {

e.preventDefault();

var dataItem = this.dataItem($(e.currentTarget).closest("tr"));

var rebuiltPartNum = dataItem.RebuiltStockNum;

// Generate the Excel file by calling SaveToExcel via AJAX

$.ajax({

url: '@Url.Action("SaveToExcel", "Export")',

type: 'GET',

data: { rebuiltPartNum: rebuiltPartNum },

xhrFields: {

responseType: 'blob' // Ensure the response is treated as a binary file

},

success: function (data) {

// Create a link element to trigger the file download

var link = document.createElement('a');

link.href = window.URL.createObjectURL(data);

link.download = `${rebuiltPartNum}\_RebuiltParts\_Estimate.xlsx`;

link.click();

// Set the subject with the rebuilt number and today's date

var today = new Date();

var subject = `Rebuilt Part Estimates for ${rebuiltPartNum} - Date ${today.toLocaleDateString()} ${today.toLocaleTimeString()}`;

// Set the form values

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

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

$("#body").val(`Dear Sir/Madam,\nPlease find attached here with Rebuilt Part Estimate for ${rebuiltPartNum}.\n\nThank you`);

// Open the Kendo Window

var kendoWindow = $("#emailEstimateWindow").data("kendoWindow");

if (kendoWindow) {

kendoWindow.center().open();

} else {

console.error("Kendo Window not found.");

}

},

error: function (xhr, status, error) {

console.error("Error downloading Excel file:", error);

}

});

}

// function onEmailEstimateClick(e) {

// e.preventDefault();

// var dataItem = this.dataItem($(e.currentTarget).closest("tr"));

// var rebuiltPartNum = dataItem.RebuiltStockNum;

// // onSaveToExcelClick(e);

// // Generate the Excel file by calling SaveToExcel

// window.location.href = '@Url.Action("SaveToExcel", "Export")' + '?rebuiltPartNum=' + rebuiltPartNum;

// // Set the subject with the rebuilt number and today's date

// var today = new Date();

// var subject = `Rebuilt Part Estimates for ${rebuiltPartNum} - Date ${today.toLocaleDateString()} ${today.toLocaleTimeString()}`;

// // Set the form values

// $("#to").val("");

// $("#subject").val(subject);

// $("#body").val(`Dear Sir/Madam,\nPlease find attached here with Rebuilt Part Estimate for ${rebuiltPartNum}.\n\nThank you`);

// // Open the Kendo Window

// var kendowindow = $("#emailEstimateWindow").data("kendoWindow");

// if (kendowindow) {

// kendowindow.center().open();

// } else {

// console.error("Kendo Window not found.");

// }

// }

function onCancelEmailClick() {

// Close the Kendo Window

var window = $("#emailEstimateWindow").data("kendoWindow");

window.close();

// Show a notification

var notification = $("#notification").data("kendoNotification");

notification.show("Email Request Cancelled", "info");

}

</script>

}

**ArchivedIndex.cshtml**

csharp

Copy

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@{

<div>ArchivedIndex.cshtml file opened</div>

}

**Controller (Example)**

csharp

Copy

using Microsoft.AspNetCore.Mvc;

namespace YourNamespace.Controllers

{

public class RebuiltPartsController : Controller

{

public IActionResult Index()

{

return View();

}

public IActionResult ArchivedIndex()

{

return View();

}

}

}

**Explanation:**

1. **TabStrip Implementation**: The RebuiltParts/Index.cshtml file now includes a Telerik TabStrip with two tabs. The first tab contains the existing grid, and the second tab loads the ArchivedIndex.cshtml view.
2. **Exception Handling**: Added a try-catch block around the Html.Partial("ArchivedIndex") call to handle any exceptions that might occur while loading the archived parts view.
3. **Controller**: Added a simple controller action ArchivedIndex to return the ArchivedIndex.cshtml view.
4. **Comments**: Added comments to explain the changes and the purpose of each section.

This setup should provide a robust solution with proper error handling and a clear structure for managing the tabs and their content.

New chat