Below are the complete and corrected versions of the Index.cshtml and shared/\_ArchivedIndex.cshtml files. These files include the necessary fixes to ensure that the onArchiveClick function is properly defined and accessible, and all other functionality works as expected.

**Index.cshtml**

html

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) // Ensure the first tab is selected by default

.Content(@<text>

@try

{

@(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().Text(" ").IconClass("k-icon k-i-edit").HtmlAttributes(new { title = "Edit" });

command.Custom("Archive").Text(" ").IconClass("k-icon k-i-folder").Click("onArchiveClick").HtmlAttributes(new { title = "Archive" });

command.Custom("Save to Excel").Text(" ").IconClass("k-icon k-i-file-excel").Click("onSaveToExcelClick").HtmlAttributes(new { title = "Save to Excel" });

command.Custom("Email Estimate").Text(" ").IconClass("k-icon k-i-email").Click("onEmailEstimateClick").HtmlAttributes(new { title = "Email Estimate" });

}).Title("Actions").Width(200).HtmlAttributes(new { @class = "action-buttons" });

})

.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))

)

)

}

catch (Exception ex)

{

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

})

)

<!-- Placeholder for dynamically loaded email estimate popup and notification -->

<div id="email-estimate-container" style="display: none;">

@Html.Partial("\_EmailEstimatePopup")

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

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

</div>

@section Scripts {

<script>

// Define onArchiveClick in the global scope

function onArchiveClick(e) {

e.preventDefault();

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

var dataItem = grid.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").data("kendoNotification");

if (notification) {

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

} else {

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

}

},

error: function (xhr, status, error) {

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

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

if (notification) {

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

}

}

});

}

$(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");

}

// Other grid event bindings...

});

</script>

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

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

</script>

}

Run HTML

**shared/\_ArchivedIndex.cshtml**

html

Copy

@using BCES.Models.Parts

@using BCES.Models.Common

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

@{

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

}

<h4>Archived Parts</h4>

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

.Name("ArchivedRebuiltPartsGrid")

.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");

})

.Pageable()

.Sortable()

.ClientDetailTemplateId("archivedTemplate")

.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(false);

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

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

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

})

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

)

)

@section Scripts {

<script>

$(document).ready(function () {

var grid = $("#ArchivedRebuiltPartsGrid").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");

}

});

</script>

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

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

</script>

}

Run HTML

**Key Fixes and Features**

1. **Global Scope for onArchiveClick**:
   * The onArchiveClick function is defined in the global scope in Index.cshtml, ensuring it is accessible to the grid's Click event handler.
2. **Notification Initialization**:
   * The Kendo Notification is initialized before it is used in the onArchiveClick function to avoid errors.
3. **Error Handling**:
   * Added error handling for the notification system to ensure it doesn't fail silently.
4. **Read-Only Grid**:
   * The ArchivedRebuiltPartsGrid in \_ArchivedIndex.cshtml is configured as read-only by setting the Editable property of the model fields to false.

**Testing**

1. Load the page and ensure no errors are displayed in the browser console.
2. Test the "Archive" button in the RebuiltPartsGrid to ensure the onArchiveClick function is called and the notification is displayed.
3. Verify that the ArchivedRebuiltPartsGrid is read-only and displays data correctly.

If you encounter any issues, let me know!