Here are the full file implementations with the requested changes, including exception handling and additional comments: ### BusesModel.cs

csharp

using System;

public class BusesModel

{

public int ListId { get; set; }

public string Description { get; set; } // Should be displayed in the column

}

### RebuiltPartsViewModel.cs

csharp

using System.Collections.Generic;

public class RebuiltPartsViewModel

{

public string RebuiltStockNum { get; set; }

// Correcting the syntax here; replace CategoryIds with a specific type if needed

// public List CategoryIds { get; set; } // Uncomment and specify type for category IDs, if needed

public List Buses { get; set; } // Correct syntax to hold a list of BusesModel

}

### ProductsController.cs Implement the controller with the action to fetch vehicle series using Dapper, adding exception handling.

csharp

using Microsoft.AspNetCore.Mvc;

using Dapper;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

public class ProductsController : Controller

{

private readonly string \_connectionString = "YourConnectionString"; // Update with your connection string

public IActionResult Index()

{

return View();

}

[HttpGet]

public IActionResult GetVehicleSeries()

{

try

{

using (var connection = new SqlConnection(\_connectionString))

{

// Fetching bus descriptions from a hypothetical `Buses` table

var buses = connection.Query("SELECT ListId, Description FROM Buses").ToList();

return Json(buses);

}

}

catch (Exception ex)

{

// Handle exceptions and log them if necessary

return StatusCode(500, $"Internal server error: {ex.Message}");

}

}

}

### Index.cshtml

@model List<RebuiltPartsViewModel>

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

.Name("RebuiltPartsGrid")

.Columns(columns =>

{

columns.Bound(c => c.RebuiltStockNum).Title("Rebuilt Stock Code");

// Add MultiSelect bus selection column here

columns.Template(p =>

{

@\* MultiSelect implementation \*@

@Html.Kendo().MultiSelect()

.Name("Buses\_" + p.RebuiltStockNum)

.DataTextField("Description")

.DataValueField("ListId")

.HtmlAttributes(new { style = "width: 100%", @class = "buses-multiselect" })

.BindTo(new List<BusesModel>()); // Placeholder; will be populated via AJAX

}).Title("Bus Selection");

})

.Pageable()

.Sortable()

.Filterable()

.DataSource(dataSource => dataSource

.Ajax()

.Read(read => read.Action("Index", "Products")) // Adjust the action if necessary

)

)

@section scripts {

<script>

$(document).ready(function () {

// For each row, call the server to get the bus list and bind it to the MultiSelect

$(".buses-multiselect").each(function (index, element) {

$.get("@Url.Action("GetVehicleSeries", "Products")", function (data) {

$(element).data("kendoMultiSelect").setDataSource(data);

});

});

});

</script>

}