I have one idea. in SupplierOrders - Create. when I select a supplier, is there any possibility to populate "Products" menu in the items with only products for the selected supplier?

Thought for 10s

Yes. Do it with a tiny JSON endpoint + a few lines of JS.

**1) Controller endpoint**

In **SupplierOrdersController**:

using Microsoft.AspNetCore.Mvc.Rendering;

[HttpGet]

public async Task<IActionResult> ProductsBySupplier(Guid supplierId, CancellationToken ct)

{

var products = await \_productService.GetBySupplierAsync(supplierId, ct); // returns ProductDto list

var options = products.Select(p => new SelectListItem

{

Value = p.ProductId.ToString(),

Text = string.IsNullOrWhiteSpace(p.Name) ? p.Code : $"{p.Code} - {p.Name}"

});

return Json(options);

}

**2) View wiring (SupplierOrders/Create.cshtml)**

Give the supplier <select> an id and add JS:

<select asp-for="SupplierId" asp-items="ViewBag.Suppliers" class="form-select" id="SupplierId"></select>

In @section Scripts:

@section Scripts {

<partial name="\_ValidationScriptsPartial" />

<script>

document.addEventListener('DOMContentLoaded', function () {

const itemsContainer = document.getElementById('items-container');

const supplierSel = document.getElementById('SupplierId');

const productsUrl = '@Url.Action("ProductsBySupplier","SupplierOrders")';

let rowIndex = 1; // Items[0] exists

// initial options from server-rendered first select

let productsOptionsHtml = (itemsContainer.querySelector('select[name="Items[0].ProductId"]')?.innerHTML) || "";

// helper: replace options for ALL product selects

function refreshAllProductSelects(newOptionsHtml) {

itemsContainer.querySelectorAll('select[name$=".ProductId"]').forEach(sel => {

const prev = sel.value; // try to keep current selection if still present

sel.innerHTML = newOptionsHtml;

if (prev && Array.from(sel.options).some(o => o.value === prev)) {

sel.value = prev;

} else {

sel.value = ""; // fall back to default

}

});

}

// on supplier change -> fetch products

supplierSel.addEventListener('change', async function () {

const supId = this.value;

if (!supId) {

productsOptionsHtml = '<option value="">-- select product --</option>';

refreshAllProductSelects(productsOptionsHtml);

return;

}

const resp = await fetch(`${productsUrl}?supplierId=${encodeURIComponent(supId)}`, { method: 'GET' });

const data = await resp.json(); // [{value, text}, ...]

const opts = ['<option value="">-- select product --</option>']

.concat(data.map(x => `<option value="${x.value}">${x.text}</option>`))

.join('');

productsOptionsHtml = opts;

refreshAllProductSelects(productsOptionsHtml);

});

// Add Item button uses the current productsOptionsHtml

window.addItemRow = function () {

const div = document.createElement('div');

div.className = 'row g-2 justify-content-between mb-2';

div.innerHTML = `

<div class="col-md-6 d-flex me-4">

<label class="form-label small mt-2 me-2">Product</label>

<select name="Items[${rowIndex}].ProductId" class="form-select">

${productsOptionsHtml}

</select>

</div>

<div class="col-md-3 d-flex">

<label class="form-label small mt-2 me-2">Quantity</label>

<input name="Items[${rowIndex}].Quantity" type="number" min="1" value="1" class="form-control" />

</div>

<div class="col-md-2 d-grid">

<button type="button" class="btn btn-outline-danger" onclick="this.closest('.row').remove()">Remove</button>

</div>`;

itemsContainer.appendChild(div);

rowIndex++;

};

});

</script>

}

That’s it: when a supplier is chosen, all item product dropdowns are repopulated with **only that supplier’s products**, and newly added rows use the same filtered list.