**Batch-Autohide:**

@page "/Demo/Batch/AutoHide/Details"

@page "/Demo/Batch/AutoHide/Details/{MasterId:int}"

@inherits MasterDetailPage<WarehouseReceipt, ReceiptDetail>

<div class="col-md-4 row pe-3">

<**CaspianForm** @ref="Form" **Model**="UpsertData">

<div class="row c-controls ps-3 pe-3">

<div class="col-md-4">

<label>Receipt date</label>

</div>

<div class="col-md-8">

<**DatePicker** @bind-Value="UpsertData.Date" />

</div>

<div class="col-md-4">

<label>Warehouse</label>

</div>

<div class="col-md-8">

<**ComboBox** **TValue**="int" **TEntity**="Warehouse" @bind-Value="UpsertData.WarehouseId" **TextExpression**="t => t.Title" />

</div>

<div class="col-md-4">

<label>Description</label>

</div>

<div class="col-md-8">

<**StringTextBox** @bind-Value="UpsertData.Comment" **MultiLine** **Rows**="3" />

</div>

<div class="col-md-12">

<**FormCommandButton** />

</div>

</div>

</**CaspianForm**>

</div>

<div class="col-md-8 ps-3">

<**DataGrid** @ref="Grid" **TEntity**="ReceiptDetail" **TableHeight**="300" **Inline** **Batch** **AutoHide** **ConditionExpr**="t => t.ReceiptId == MasterId">

<**Columns** **Context**="con">

<**GridColumn** **Field**="t => t.Material.Title" **Title**="Material">

<**EditTemplate**>

<**AutoComplete** **TValue**="int" **TEntity**="Material" @bind-Value="con.Data.MaterialId"

**HideHeader** **AutoHide** **Title**="Material" **TextExpression**="t => t.Title"

**OnChange**="async () => await UpdateMaterialReceipt(con.Data.MaterialId, con.Data)">

<**MaterialLookupWindow** />

</**AutoComplete**>

</**EditTemplate**>

</**GridColumn**>

<**GridColumn** **Field**="t => t.QuantityMain" style="width:110px;">

<**EditTemplate**>

<**NumericTextBox** @ref="txtMainUnit" @bind-Value="con.Data.QuantityMain"

@onkeyup="async e => await MainUnitChanged(e, con.Data, con.UpsertMode.Value)" />

</**EditTemplate**>

</**GridColumn**>

<**GridColumn** **Field**="t => t.Material.MainUnit.Title" **Title**="Main unit" style="width:80px">

<**Template**>

<span>@(con.Data?.Material?.MainUnit?.Title)</span>

</**Template**>

</**GridColumn**>

<**GridColumn** **Field**="t => t.QuantitySub" style="width:110px;">

<**EditTemplate**>

@if (con.Data?.Material?.SubunitId != null)

{

<**NumericTextBox** @ref="txtSubunit" @bind-Value="con.Data.QuantitySub"

@onkeyup="async e => await SubUnitChanged(e, con.Data, con.UpsertMode.Value)" />

}

</**EditTemplate**>

</**GridColumn**>

<**GridColumn** **Field**="t => t.Material.Subunit.Title" **Title**="Sub unit" style="width:80px">

<**Template**>

<span>@(con.Data?.Material?.Subunit?.Title)</span>

</**Template**>

</**GridColumn**>

<**GridCommandColumns** />

</**Columns**>

</**DataGrid**>

</div>

@code

{

NumericTextBox<int> txtMainUnit;

NumericTextBox<int?> txtSubunit;

protected override Task OnMasterEntityCreatedAsync()

{

if (UpsertData.Id == 0)

UpsertData.Date = DateTime.Now.Date;

return base.OnMasterEntityCreatedAsync();

}

async Task MainUnitChanged(KeyboardEventArgs e, ReceiptDetail detail, UpsertMode mode)

{

if (e.Key == "Enter" || e.Key == "NumpadEnter")

{

if (detail?.Material.SubunitId != null)

await txtSubunit.FocusAsync();

else

await Grid.ValidateAndUpsert(mode);

}

}

async Task SubUnitChanged(KeyboardEventArgs e, ReceiptDetail detail, UpsertMode mode)

{

if (e.Key == "Enter" || e.Key == "NumpadEnter")

await Grid.ValidateAndUpsert(mode);

}

async Task UpdateMaterialReceipt(int materialId, ReceiptDetail detail)

{

if (materialId > 0)

{

using var scope = CreateScope();

detail.Material = await new MaterialService(scope.ServiceProvider).GetAll().Where(t => t.Id == materialId)

.Include(t => t.MainUnit).Include(t => t.Subunit).SingleAsync();

await txtMainUnit.FocusAsync();

}

}

}