**Search*:***

<div class="col-md-6">

<div>

<**StringTextBox** @bind-Value="@strSearch" placeholder="City search" **BindingType**="BindingType.OnInput"

**OnChange**="SearchOnTree" />

</div>

<div>

<**TreeView** **TEntity**="TreeViewItem" style="height:500px;" **Source**="searchSource" />

</div>

</div>

@code

{

string strSearch;

IList<TreeViewItem> searchSource;

//-----Search Method ---

async Task SearchOnTree()

{

using var scope = CreateScope();

var query = new CountryService(scope.ServiceProvider).GetAll();

if (strSearch.HasValue())

query = query.Where(t => t.Cities.Any(u => u.Title.Contains(strSearch)));

searchSource = await query.Select(t => new TreeViewItem()

{

Collabsable = true,

Expanded = true,

Text = t.Title,

Value = t.Id.ToString(),

Items = t.Cities.Where(t => strSearch == null || t.Title.Contains(strSearch)).Select(u => new

TreeViewItem()

{

Collabsable = false,

Text = u.Title,

Value = u.Id.ToString()

}).ToList()

}).ToListAsync();

await base.OnInitializedAsync();

}

}

**Selection*:***

<div class="c-controls ps-4 pe-4 pb-2">

<label>You selected @(selecteCitiesCount) city</label>

</div>

<**TreeView** @ref="tree" **TEntity**="TreeViewItem" style="height:500px;" **Source**="selectionSource"

**OnChange**="UpdateSelecting" />

@code

{

TreeView<TreeViewItem> tree;

IList<TreeViewItem> selectionSource;

protected override async Task OnInitializedAsync()

{

using var scope = CreateScope();

var query = new CountryService(scope.ServiceProvider).GetAll();

selectionSource = await query.Select(t => new TreeViewItem()

{

Value = t.Id.ToString(),

Text = t.Title,

Expanded = true,

Selectable = true,

Items = t.Cities.Select(u => new TreeViewItem()

{

Value = u.Id.ToString(),

Text = u.Title,

Collabsable = false,

Selectable = true,

}).ToList()

}).ToListAsync();

await base.OnInitializedAsync();

}

void UpdateSelecting(TreeViewItem node)

{

///For a two-level tree

///first level

if (node.Depth == 1)

{

foreach (var child in node.Items)

child.Selected = node.Selected;

}

///second level

if (node.Depth == 2)

{

var siblingNodes = node.Parent.Items;

if (siblingNodes.All(t => t.Selected == true))

node.Parent.Selected = true;

else if (siblingNodes.All(t => t.Selected == false))

node.Parent.Selected = false;

else

node.Parent.Selected = null;

}

selecteCitiesCount = tree.GetSeletcedItems().Count(t => t.Depth == 2);

}

}