To implement Telerik ComboBox for Supplier Number and Name with AutoComplete logic and make them read-only, follow these steps:

**1. Create a ComboBox Editor Template (ComboBoxEditor.cshtml):**

html

Copy

@model object

@{

var fieldName = ViewData.TemplateInfo.GetFullHtmlFieldName("");

}

@(Html.Kendo().ComboBox()

.Name(fieldName)

.DataTextField("Text")

.DataValueField("Value")

.Filter(FilterType.Contains)

.Placeholder("Select...")

.HtmlAttributes(new { style = "width:100%;", @class = "readonly-combobox" })

.ReadOnly(true) // Makes the ComboBox input read-only (can't type)

.DataSource(source =>

{

source.Read(read =>

{

read.Url(Url.Action("GetAutoCompleteData", "AutoComplete"))

.Type(HttpVerbs.Get)

.Data("getAutoCompleteParams");

})

.ServerFiltering(true);

})

)

Run HTML

**2. Update Grid Columns in Index.cshtml:**

csharp

Copy

columns.Bound(p => p.OrigSuppNum)

.Title("Supplier Number")

.Filterable(ftb => ftb.Cell(cell => cell.Operator("contains")))

.EditorTemplateName("ComboBoxEditor"); // Use ComboBox template

columns.Bound(p => p.OrigSupplierName)

.Title("Supplier Name")

.Filterable(ftb => ftb.Cell(cell => cell.Operator("contains")))

.EditorTemplateName("ComboBoxEditor"); // Use ComboBox template

**3. Modify getAutoCompleteParams JavaScript Function:**

Update the function to target both ComboBox and AutoComplete inputs:

javascript

Copy

function getAutoCompleteParams() {

// Target both AutoComplete and ComboBox inputs

var input = $("input[data-role='autocomplete'], input[data-role='combobox']").filter(":focus");

return {

columnName: input.attr("name"),

searchText: input.val() || ""

};

}

**4. CSS to Style Read-Only ComboBox (Optional):**

Add to your styles to visually indicate read-only state:

css

Copy

.k-readonly .k-input {

background-color: #f5f5f5;

cursor: not-allowed;

}

**5. Ensure Controller Compatibility:**

The AutoCompleteController should handle requests for OrigSuppNum and OrigSupplierName columns. Example:

csharp

Copy

public class AutoCompleteController : Controller

{

[HttpGet]

public JsonResult GetAutoCompleteData(string columnName, string searchText)

{

// Logic to fetch data based on columnName

var data = new List<dynamic>();

if (columnName == "OrigSuppNum")

{

data = FetchSupplierNumbers(searchText);

}

else if (columnName == "OrigSupplierName")

{

data = FetchSupplierNames(searchText);

}

return Json(data.Select(x => new { Text = x, Value = x }));

}

}