Got it! Let’s address all the issues systematically and fix the problems:

1. **Move onAutocompleteData to the main view (Index.cshtml)**, as you mentioned it throws an error when placed in the editor template.
2. Fix ViewData["ColumnName"] being null when the CommonAutoComplete editor template is called.
3. Dynamically pass the controller name via ViewData.
4. Debug the text parameter issue step-by-step using Chrome Developer Tools.
5. Provide the full and corrected files.

Here’s the updated and complete solution:

**1. Fixing the AutoComplete Data Passing**

To dynamically handle columnName and text and make it work across columns, we need:

* A way to pass the columnName to the editor template correctly.
* Ensure onAutocompleteData works globally for all AutoComplete fields.

**Updated Grid Definition (Index.cshtml)**

razor

CopyEdit

@{

ViewData["ControllerName"] = "Grid"; // Pass the controller name dynamically

}

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

.Name("Grid")

.Columns(columns =>

{

columns.Bound(c => c.StockCode)

.Title("Stock Code")

.EditorTemplateName("CommonAutoComplete") // Shared autocomplete editor

.Width(150);

columns.Bound(c => c.Description)

.Title("Description")

.EditorTemplateName("CommonAutoComplete") // Shared autocomplete editor

.Width(200);

columns.Bound(c => c.TotalCost)

.Title("Total Cost")

.EditorTemplateName("CommonAutoComplete") // Shared autocomplete editor

.Width(150);

columns.Command(command => command.Edit()).Width(150); // Inline edit button

})

.Editable(editable => editable.Mode(GridEditMode.InLine)) // Inline editing mode

.Pageable()

.Sortable()

.DataSource(dataSource => dataSource

.Ajax()

.Read(read => read.Action("GetGridData", (string)ViewData["ControllerName"])) // Dynamic controller name

.Update(update => update.Action("UpdateGridData", (string)ViewData["ControllerName"]))

)

)

<script>

// Global JavaScript function to pass both columnName and text

function onAutocompleteData() {

var grid = $("#Grid").data("kendoGrid"); // Get the grid instance

var columnName = grid.currentDataItem(); // Get the current column being edited

return {

columnName: columnName.field, // Dynamically pass the column name

text: this.value // Pass the user's input text

};

}

</script>

**Updated Editor Template (CommonAutoComplete.cshtml)**

razor

CopyEdit

@model object

@(Html.Kendo().AutoComplete()

.Name(ViewContext.ViewData["ColumnName"]?.ToString() ?? "UnknownField") // Dynamically set the name

.Filter("contains") // Filter type

.DataTextField("Value") // Field name of the result returned from the controller

.MinLength(2) // Start searching after 2 characters

.DataSource(source => source

.Read(read => read.Action("GetColumnSuggestions", (string)ViewData["ControllerName"])

.Data("onAutocompleteData")) // Pass columnName and text dynamically

)

.HtmlAttributes(new { @class = "k-input" })

)

**2. Updated Controller Logic**

We need to dynamically fetch data for the column specified in columnName and based on the user input (text).

**Controller: GridController.cs**

csharp

CopyEdit

using Dapper;

using Microsoft.AspNetCore.Mvc;

using System.Data;

using System.Data.SqlClient;

public class GridController : Controller

{

private readonly string \_connectionString = "YourDatabaseConnectionString";

// Action to provide grid data

public IActionResult GetGridData()

{

// Mock data for the grid

var data = new[]

{

new StockModel { StockCode = 101, Description = "Item 101", TotalCost = 150.50M },

new StockModel { StockCode = 102, Description = "Item 102", TotalCost = 250.75M }

};

return Json(data);

}

// Action to update grid data

[HttpPost]

public IActionResult UpdateGridData([FromBody] StockModel model)

{

// Update logic here (e.g., update database using Dapper)

return Json(model);

}

// Action to fetch autocomplete suggestions

public IActionResult GetColumnSuggestions(string columnName, string text)

{

if (string.IsNullOrWhiteSpace(columnName) || string.IsNullOrWhiteSpace(text))

{

return Json(new List<object>()); // Return an empty list if input is invalid

}

using (IDbConnection db = new SqlConnection(\_connectionString))

{

// Dynamically select the query based on the column name

string query = columnName switch

{

"StockCode" => "SELECT DISTINCT StockCode AS Value FROM StockCodes WHERE CAST(StockCode AS NVARCHAR) LIKE @SearchText",

"Description" => "SELECT DISTINCT Description AS Value FROM StockDescriptions WHERE Description LIKE @SearchText",

"TotalCost" => "SELECT DISTINCT TotalCost AS Value FROM StockCosts WHERE CAST(TotalCost AS NVARCHAR) LIKE @SearchText",

\_ => throw new ArgumentException("Invalid column name.")

};

// Use Dapper to execute the query and fetch results

var parameters = new { SearchText = "%" + text + "%" };

var result = db.Query<object>(query, parameters).ToList();

return Json(result);

}

}

}

**3. Debugging with Chrome Developer Tools**

If text is still null, here’s how to debug it using Chrome Developer Tools:

1. **Open Developer Tools**:
   * Right-click on the page and select **Inspect**.
   * Go to the **Network** tab.
2. **Trigger the Request**:
   * Open the inline editor in the grid by clicking the **Edit** button.
   * Type some text into the AutoComplete field (e.g., for StockCode).
3. **Find the Request**:
   * Look for the AJAX request to GetColumnSuggestions.
   * It should appear in the list of network requests.
4. **Inspect the Request**:
   * Click on the request.
   * Go to the **Headers** tab.
   * Look under **Query String Parameters** or **Request Payload** to see if text and columnName are being sent.
5. **Inspect the Response**:
   * Go to the **Response** tab to see the data returned by the server.

**4. Fixing Null ViewData["ColumnName"]**

If ViewData["ColumnName"] is null, make sure to set the column name dynamically in the grid configuration:

**Updated Column Definition in Index.cshtml**

razor

CopyEdit

columns.Bound(c => c.StockCode)

.Title("Stock Code")

.EditorTemplateName("CommonAutoComplete")

.ClientTemplate("<input type='hidden' value='StockCode' name='ColumnName' />");

You can pass it dynamically by setting the correct value for the column.

**5. Full Example**

Let me know if the text issue persists after debugging with Chrome tools! I'll guide you further based on the findings.