**Step 1: Update AutocompleteEditor.cshtml (Key Fix)**

html

Copy

@model object

@{

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

}

@(Html.Kendo().AutoComplete()

.Name(fieldName)

.DataTextField("this")

.Filter("contains")

.Placeholder("Type to search...")

.HtmlAttributes(new { data\_column\_name = fieldName }) <!-- Store column name -->

.DataSource(source =>

{

source.Read(read =>

{

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

.Type(HttpVerbs.Get)

.Data("getAutoCompleteParams"); <!-- Use named function -->

})

.ServerFiltering(true);

})

)

Run HTML

**Step 2: Add Global JavaScript Function (in LabourTask.cshtml or layout)**

javascript

Copy

<script>

function getAutoCompleteParams() {

// Get the AutoComplete input triggering the request

var autoComplete = this.element.closest("[data-role=autocomplete]").data("kendoAutoComplete");

return {

columnName: autoComplete.element.attr("name"), // Dynamic column name

searchText: autoComplete.value() // Current typed value

};

}

</script>

**Step 3: Verify Controller Action (AutoCompleteController.cs)**

csharp

Copy

[HttpGet]

public IActionResult GetAutoCompleteData(string columnName, string searchText)

{

// Parameter validation

if (string.IsNullOrEmpty(columnName))

return BadRequest("Column name is required.");

// Column mapping (add new entries here for other columns)

var columnMappings = new Dictionary<string, (string SqlColumn, string Table)>

{

{ "TaskDescription", ("TaskDescription", "[SBCES].[LabourTaskDescriptions]") }

};

if (!columnMappings.TryGetValue(columnName, out var mapping))

return BadRequest("Invalid column name.");

// Parameterized query

using (var connection = \_dbConnection)

{

var sql = $@"

SELECT DISTINCT {mapping.SqlColumn}

FROM {mapping.Table}

WHERE {mapping.SqlColumn} LIKE @SearchText + '%'";

var results = connection.Query<string>(sql, new { SearchText = searchText ?? "" });

return Json(results.ToList()); // Return direct array

}

}

**Step 4: Update Grid Column Configuration**

csharp

Copy

columns.Bound(p => p.TaskDescription)

.Title("Task Description")

.EditorTemplateName("AutoCompleteEditor"); // Must match template name

**Key Changes Explained:**

1. **Data Attribute Binding**: The AutoComplete now stores its column name in data-column-name.
2. **Parameter Function**: getAutoCompleteParams explicitly sends columnName and searchText.
3. **Direct Array Response**: Controller returns a simple string array (["Item1","Item2"]).

**Testing:**

1. **Network Request**: Verify requests now have:

Copy

/AutoComplete/GetAutoCompleteData?columnName=TaskDescription&searchText=ad

1. **Response Format**: Ensure the response is a JSON array like ["Task 1","Task 2"].