**1. Updated Controller Method**

csharp

Copy

[HttpGet]

public async Task<IActionResult> GetAutoCompleteData(

string columnName,

string searchText,

int page = 1)

{

// Validate columnName first

if (string.IsNullOrWhiteSpace(columnName))

{

return BadRequest("Column name is required");

}

// Add null check for dictionary lookup

if (!\_columnMappings.ContainsKey(columnName))

{

return BadRequest($"Invalid column name: {columnName}");

}

if (!\_columnMappings.TryGetValue(columnName, out var mapping))

{

return BadRequest($"Column mapping failed for: {columnName}");

}

// Rest of your code...

}

**2. Enhanced JavaScript**

javascript

Copy

function getDropDownListParams() {

try {

const grid = $("#NscPartsUsedGrid").data("kendoGrid");

if (!grid) {

console.error("Grid not found");

return {};

}

const editContainer = grid.editContainer;

if (!editContainer) {

console.warn("No active edit container");

return {};

}

const dropdown = $(editContainer).find("input[data-role='dropdownlist']").data("kendoDropDownList");

if (!dropdown) {

console.warn("Dropdown not found in edit container");

return {};

}

const columnName = dropdown.element.attr("data-column") || "";

const searchText = dropdown.filterInput.val() || "";

console.log("Params:", { columnName, searchText, page: dropdown.dataSource.page() });

return {

columnName: columnName,

searchText: searchText,

page: dropdown.dataSource.page() || 1

};

} catch (e) {

console.error("Error in getDropDownListParams:", e);

return {};

}

}

**3. Verify Dropdown Template**

Ensure your DropDownList template includes the data-column attribute:

html

Copy

@(Html.Kendo().DropDownList()

.Name(fieldName)

.HtmlAttributes(new {

data\_role = "dropdownlist",

data\_column = fieldName // This must match your grid column field

}))

Run HTML

**Key Fixes:**