**Updated JavaScript Parameter Function**

javascript

Copy

function getDropDownListParams() {

// Get grid reference

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

if (!grid) {

console.warn("Grid not found");

return {};

}

// Find current edit cell

const editCell = grid.element.find(".k-edit-cell");

if (editCell.length === 0) {

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

return {};

}

// Find dropdown in edit cell

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

if (!dropdown) {

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

return {};

}

// Get parameters

return {

columnName: dropdown.element.attr("data-column") || "",

searchText: dropdown.filterInput.val() || "",

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

pageSize: dropdown.dataSource.pageSize() || 20

};

}

**2. Controller Modifications**

csharp

Copy

[HttpGet]

public async Task<IActionResult> GetAutoCompleteData(

[FromQuery] string columnName,

[FromQuery] string searchText,

[FromQuery] int page = 1,

[FromQuery] int pageSize = 20)

{

// Add null coalescing for parameters

columnName ??= string.Empty;

searchText ??= string.Empty;

// Rest of your existing validation

if (string.IsNullOrEmpty(columnName))

return BadRequest("Column name is required");

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

return BadRequest("Invalid column name");

// Modified query with parameter fixes

var sql = $@"SELECT DISTINCT {mapping.Value} AS Value,

{mapping.Text} AS Text

FROM {mapping.Table}

WHERE {mapping.Text} LIKE @SearchText

ORDER BY {mapping.Text}

OFFSET @Offset ROWS

FETCH NEXT @PageSize ROWS ONLY";

var results = await \_dbConnection.QueryAsync<dynamic>(sql, new {

SearchText = $"{searchText}%", // Changed to starts-with for better perf

Offset = (page - 1) \* pageSize,

PageSize = pageSize

});

return Json(results);

}

**3. DropDownList Configuration Fix**

csharp

Copy

.HtmlAttributes(new {

data\_role = "dropdownlist",

data\_column = fieldName, // Ensure this matches grid column field

name = fieldName // Required for parameter binding

})

**Key Changes:**

1. **Explicit Parameter Binding**  
   Added [FromQuery] attributes to ensure proper model binding
2. **Null Coalescing**  
   Handles null parameters at the method entry point
3. **Direct DOM Selection**  
   No longer relies on global variables, finds elements fresh each request
4. **Starts-With Filter**  
   Changed LIKE '%...' to LIKE '...%' for better index utilization

**Verification Steps:**

1. **Check Network Requests**  
   Look for parameters in the request URL:  
   ?columnName=...&searchText=...&page=...&pageSize=...
2. **Add Console Logging**  
   In the parameter function:

javascript

Copy

console.log("Submitting params:", params);

1. **Validate HTML Attributes**  
   Ensure dropdown elements have:

html

Copy

<input data-role="dropdownlist"

data-column="YourColumnName"

name="YourColumnName">

Run HTML

1. **Test With Simple Column**  
   Start with a basic column mapping to verify the pipeline

**Why This Works:**

* **Direct DOM Access** avoids timing issues with grid edit events
* **Explicit Query Attributes** ensure parameter binding
* **Fresh Element Selection** guarantees current state
* **Parameter Validation** happens before dictionary lookup

This maintains the performance improvements while restoring the original parameter-passing functionality. The null parameters should now be resolved with proper element selection and parameter binding.

New chat