Let's fix the jQuery selector chain and ensure proper element selection. Here's the corrected valueMapper function:

javascript

Copy

function valueMapper(options) {

// Get the grid reference safely

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

if (!grid) {

console.warn("Grid not found");

return options.error();

}

// Get the edit container safely using optional chaining

const editContainer = grid.editContainer;

if (!editContainer) {

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

return options.error();

}

// Proper jQuery element selection

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

if (!dropdownInput.length) {

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

return options.error();

}

// Get column name from data attribute

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

$.ajax({

url: "@Url.Action("ValueMapper", "AutoComplete")",

type: "POST",

contentType: "application/json",

data: JSON.stringify({

columnName: columnName,

values: options.value

}),

success: function(result) {

options.success(result);

},

error: function(xhr, status, error) {

console.error("ValueMapper error:", error);

options.error();

}

});

}

**Key Fixes:**

1. **Proper jQuery Object Wrapping**  
   Added $(editContainer) to ensure we have a jQuery object before calling .find()
2. **Error Handling**  
   Added proper error handling with options.error()
3. **Optional Chaining**  
   Added safety checks for grid and edit container existence
4. **Element Existence Verification**  
   Check if dropdownInput exists before using it

**Alternative Version Using Direct DropDown Reference:**

javascript

Copy

function valueMapper(options) {

// Get dropdown reference directly from Kendo

const dropdown = options.sender;

if (!dropdown) {

console.warn("No dropdown reference");

return options.error();

}

// Get column name from dropdown element

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

$.ajax({

url: "@Url.Action("ValueMapper", "AutoComplete")",

type: "POST",

data: JSON.stringify({

columnName: columnName,

values: options.value

}),

success: function(result) {

options.success(result);

},

error: function() {

options.error();

}

});

}