1. **AutoComplete Widget Not Initializing**: Caused by incorrect data binding or initialization timing.
2. **e.slice Error**: Typically occurs when the AutoComplete receives invalid data (e.g., null or non-array responses).

**Step-by-Step Fixes**

**1. Update AutocompleteEditor.cshtml (Critical Fix for Data Binding):**

html

Copy

@model object

@{

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

}

@(Html.Kendo().AutoComplete()

.Name(fieldName)

.DataTextField("this") // <-- KEY CHANGE: "this" for string values

.Filter("contains")

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

.HtmlAttributes(new { data\_editor\_type = "autocomplete" })

.DataSource(source =>

{

source.Read(read =>

{

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

.Type(HttpVerbs.Get)

.Data("getAdditionalData"); // <-- Use a named function

})

.ServerFiltering(true);

})

)

Run HTML

**2. Fix the onAdditionalData Function (LabourTask.cshtml):**

javascript

Copy

<script>

function getAdditionalData(e) {

var input = $("input[name='" + e.field + "']"); // Find input by field name

return {

columnName: e.field, // Pass the column name dynamically

searchText: input.val() // Get current input value

};

}

function onGridEdit(e) {

// Check all AutoComplete inputs in the edit container

e.container.find("input[data-role='autocomplete']").each(function() {

var input = $(this);

var widget = input.data("kendoAutoComplete");

if (!widget) {

console.error("AutoComplete widget not found for:", input.attr("name"));

} else {

console.log("AutoComplete initialized for:", input.attr("name"));

}

});

}

</script>

**3. Fix the Controller Response (AutoCompleteController.cs):**

csharp

Copy

[HttpGet]

public IActionResult GetAutoCompleteData(string columnName, string searchText)

{

// ... [existing validation logic] ...

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 array directly (not wrapped in "data")

}

}

**4. Ensure Grid Column Configuration:**

csharp

Copy

columns.Bound(p => p.TaskDescription)

.Title("Task Description")

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

**Key Explanations**

1. **DataTextField("this")**:
   * When your data source returns simple strings (not objects), "this" tells Kendo to bind directly to the string value. Using "Text" caused binding failures.
2. **Controller Response**:
   * The AutoComplete expects a flat array of strings (e.g., ["Item1", "Item2"]). Returning new { data = results } wrapped the array in an unnecessary object, causing the e.slice error.
3. **getAdditionalData**:
   * Explicitly passes the column name and current input value. Avoid using this context inside the data function to prevent scope issues.

**Testing Checklist**

1. **Network Tab**:
   * When typing in the AutoComplete, verify that requests are sent to /AutoComplete/GetAutoCompleteData with parameters:

Copy

columnName: "TaskDescription"

searchText: "a" (whatever you type)

1. **Response Format**:
   * Ensure the response is a JSON array (e.g., ["Task 1","Task 2"]), not an object like { data: [...] }.
2. **Console Errors**:
   * Confirm no errors like *"Uncaught TypeError: e.slice is not a function"* remain.