**1. Hide ID Column in Grid View**

**Modify Index.cshtml** (add .Hidden(true) to ID column):

csharp

Copy

columns.Bound(p => p.NscPartsUsedId)

.Title("Part Id")

.Hidden(true); // Add this line to hide the column

**2. Add Kendo Notifications**

**Add Notification Container** at the top of Index.cshtml:

html

Copy

<div id="notification"></div>

Run HTML

**Add Notification Script** in the Scripts section:

javascript

Copy

<script>

var notification = $("#notification").kendoNotification({

position: {

pinned: true,

top: 50,

right: 30

},

autoHideAfter: 4000,

stacking: "down",

templates: [{

type: "success",

template: "<div style='padding: 10px; background-color: #dff0d8; color: #3c763d; border: 1px solid #d6e9c6;'>#= message #</div>"

}, {

type: "error",

template: "<div style='padding: 10px; background-color: #f2dede; color: #a94442; border: 1px solid #ebccd1;'>#= message #</div>"

}]

}).data("kendoNotification");

</script>

**Modify DataSource Configuration** to handle notifications:

csharp

Copy

.DataSource(dataSource => dataSource

.Ajax()

.Events(events => events

.Error("onDataSourceError")

.RequestEnd("onRequestEnd")

)

// rest of existing configuration

)

**Add These JavaScript Functions**:

javascript

Copy

function onDataSourceError(e) {

if (e.errors) {

notification.show({

message: e.errors.message || "An unexpected error occurred",

type: "error"

});

}

}

function onRequestEnd(e) {

if (e.type !== "read") {

var message = "";

var type = "success";

if (e.response && e.response.error) {

message = e.response.error;

type = "error";

} else {

switch (e.type) {

case "create":

message = "Record created successfully";

break;

case "update":

message = "Record updated successfully";

break;

case "destroy":

message = "Record deleted successfully";

break;

}

}

if (message) {

notification.show({

message: message,

type: type

});

}

}

}

**Important Notes:**

1. **No Controller Changes Needed**: The existing error handling in your controller will automatically work with this notification system
2. **Security**: The notifications will show exactly what your controller returns (keep error messages user-friendly)
3. **Styling**: The notification templates include basic styling - modify the CSS in the templates as needed