

23-01 Resources view – Sorting – Interactive column sorting

This example shows how the automatic sorting function can be used according to the values of a selected column.

The sort is a display sort and is only maintained as long as the sort function is active. During this time, however, the sort is dynamic — i.e., changes to the data in the relevant column automatically update the sort order. If there are identical values, the initial order is preserved.

Only one column can be sorted at a time. When switching from one sorted column to another, the sort is applied to the new column. The starting point for sorting is the original order.

To use this functionality, two options must be set to **true**:

`interactiveSwitchingOfSortOrderEnabled: true,`
`sortingIndicatorVisible: true`

The **interactiveSwitchingOfSortOrderEnabled** option allows sorting by clicking on one of the table columns. However, only the second option, **sortingIndicatorVisible**, provides visual feedback about the sort mode by displaying an arrow: pointing up for ascending sort and down for descending sort.



Clicking in the column header (e.g. Start) causes two options to be set automatically:

```
resourceRowSortCodePropertyName: "Start",  
resourceRowSortMode: Enum.RowSortMode.Ascending
```

If certain columns are to be excluded from the sorting option, the **cancel** property must be set to **true** for these in the **onRowSortingChangeRequested** callback.

```
onRowSortingChangeRequested: function (args) {  
    if (args.sortCodeSource == "Name")  
        args.cancel = true;  
},
```

02-10 ResourcesView - Interactive table column sorting

Deactivate table column sorting

| | Resource | Start | End | Working Days | Sort Code 1 | Sort Code 2 | January 2028 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|-------------|------------|------------|--------------|-------------|-------------|--------------|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|--|--|--|--|--|
| | | | | | | | KW 52 | KW 01 | | | | | | | | KW 02 | | | | | | | | KW 03 | | | | | | | | | | |
| | | | | | | | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | | | | | |
| 1 | Resource 1 | 05.01.2028 | 08.01.2028 | 3 | A | a | | | | | | | | | A1 | | | | | | | | | | | | | | | | | | | |
| 2 | Resource 2 | 06.01.2028 | 09.01.2028 | 3 | C | b | | | | | | | | | A2 | | | | | | | | | | | | | | | | | | | |
| 3 | Resource 3 | 15.01.2028 | 18.01.2028 | 3 | C | c | | | | | | | | | | | | | | | | A3 | | | | | | | | | | | | |
| 4 | Resource 4 | 20.01.2028 | 23.01.2028 | 3 | A | c | | | | | | | | | | | | | | | | | | | | | | A4 | | | | | | |
| 5 | Resource 5 | 25.01.2028 | 28.01.2028 | 3 | B | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Resource 6 | 05.01.2028 | 08.01.2028 | 3 | B | c | | | | | | | | | A6 | | | | | | | | | | | | | | | | | | | |
| 7 | Resource 7 | 06.01.2028 | 13.01.2028 | 7 | B | d | | | | | | | | | A7 | | | | | | | | | | | | | | | | | | | |
| 8 | Resource 8 | 15.01.2028 | 18.01.2028 | 3 | B | d | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | Resource 9 | 20.01.2028 | 23.01.2028 | 3 | B | a | | | | | | | | | | | | | | | | A8 | | | | | | | | | | | | |
| 10 | Resource 10 | 25.01.2028 | 28.01.2028 | 3 | B | c | | | | | | | | | | | | | | | | | | | | | | A9 | | | | | | |

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>02-10 NETRONIC VSW SE</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <link href="../../VSWidget/nwaf-apptools.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-table.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-gantt.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-rab.min.css" rel="stylesheet"/>
  <script src="https://cdn.jsdelivr.net/npm/hammerjs@2.0.8/hammer.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/d3@7.9.0/dist/d3.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/tinycolor2@1.6.0/cjs/tinycolor.min.js"></script>
  <script src="../../VSWidget/nwaf-apptools.min.js"></script>
  <script src="../../VSWidget/nwaf-table.min.js"></script>
  <script src="../../VSWidget/nwaf-gantt.min.js"></script>
  <script src="../../VSWidget/nwaf-rab.min.js"></script>
  <script src="../../LicenseKey.js"></script>
  <script src="Miscellaneous.js"></script>
</head>
<body>
  <div id="toolbar">
    <h3>
      <a href="docs/23-01_ResourcesView-Sorting-Interactive_column_sorting.pdf"
        target="_blank">23-01 ResourcesView - Sorting - Interactive column sorting</a>
    </h3>
    <input id="sorting" type="button" value="Deactivate table column sorting"
      style="margin-left: 15px; height: 20px; width: 200px"/>
    </div>
  <div id="VSWidget"></div>
  <script>
    Miscellaneous.ready(() => {
      const { BarDragModes, BarSortMode, RowDesigns,
        RowSortMode, HorizontalAlignment, ViewType } = netronic.nVSW;

      document.getElementById("sorting").addEventListener("click", enableOrDisableSorting);

      const timeAreaStart = new Date(2028, 0, 1);
      const timeAreaEnd = new Date(2028, 3, 1);

      const options = {
        licenseKey: window.VSW_LICENSE_KEY,

        viewType: ViewType.Resources,

        start: timeAreaStart,
        end: timeAreaEnd,

        defaultResourceExpandedRowDesign: RowDesigns.Bars,
        defaultResourceCollapsedRowDesign: RowDesigns.Bars |
          RowDesigns.BarsInHiddenDescendantRows |
          RowDesigns.BarsStacked,
        allocationBarSortModeForStackedRowDesign: BarSortMode.StartAndEnd,
        defaultAllocationAllowedBarDragModes: BarDragModes.None,
        defaultAllocationBarSelectable: false,
        defaultResourceRowSelectable: false,
        multipleSelectionEnabled: false,
        defaultResourceTableRowDefinitionID: "CustomRowDef",
        intlDateTimeFormatOptionsMap: {
          en: {
            year: "numeric",
            month: "2-digit",
            day: "2-digit"
          },
        },
        interactiveSwitchingOfSortOrderEnabled: true,
        sortingIndicatorVisible: true,
        fixedTableColumnWidth: 0,

        onRowSortingChangeRequested: (args) => {
          if (args.sortCodeSource === "my_Name")
            args.cancel = true;
        },
      };
    });
  </script>

```

```

const tableRowDefinitions = createTableRowDefinitions();
const allocations = createAllocations();
const resources = createResources();

const vsWidget = Miscellaneous.instantiateVSWidget(document.querySelector("#VSWidget"), options);

vsWidget.addTableRowDefinitions(tableRowDefinitions);
vsWidget.addResources(resources);
vsWidget.addAllocations(allocations);
vsWidget.render();

function createTableRowDefinitions() {
    const tableRowDefinitions = [
        {
            ID: "CustomRowDef",
            CellDefinitions: [
                {
                    TitleText: "No.",
                    TextSource: "my_No",
                    HorizontalAlignment: HorizontalAlignment.Center,
                    Width: 50,
                },
                {
                    TitleText: "Resource",
                    TextSource: "my_Name",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    Width: 170,
                },
                {
                    TitleText: "Start",
                    TextSource: "my_Start",
                    TextFormat: "{{my_Start:date:en}}",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    BackgroundColor: "lightyellow",
                    Width: 120,
                },
                {
                    TitleText: "End",
                    TextSource: "my_EndDay",
                    TextFormat: "{{my_EndDay:date:en}}",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    BackgroundColor: "lightgreen",
                    Width: 100,
                },
                {
                    TitleText: "Working Days",
                    TextSource: "my_WorkingDays",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    Width: 120,
                },
                {
                    TitleText: "Sort Code 1",
                    TextSource: "my_SortCode1",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    Width: 120,
                },
                {
                    TitleText: "Sort Code 2",
                    TextSource: "my_SortCode2",
                    HorizontalTitleAlignment: HorizontalAlignment.Center,
                    HorizontalAlignment: HorizontalAlignment.Center,
                    Width: 120,
                },
            ],
        },
    ];

    return tableRowDefinitions;
}

```

```

function createResources() {
  const sortCode1 = ["A", "C", "C", "A", "B", "B", "B", "B", "B", "B"];
  const sortCode2 = ["a", "b", "c", "c", "b", "c", "d", "d", "a", "c"];
  const resources = [];

  for (let i = 1; i <= 10; i++) {
    const resource = {
      ID: `Res${i}`,
      my_Name: `Resource ${i}`,
      my_No: i,
      my_SortCode1: sortCode1[i - 1],
      my_SortCode2: sortCode2[i - 1],
      my_Start: allocations[i - 1].Entries[0].Start,
      my_EndDay: allocations[i - 1].Entries[0].my_EndDay,
      my_WorkingDays: allocations[i - 1].my_WorkingDays,
    };

    resources.push(resource);
  }

  return resources;
}

function createAllocations() {
  const allocations = [];
  const startDays = [5, 6, 15, 20, 25, 5, 6, 15, 20, 25];
  const workingDays = [3, 3, 3, 3, 3, 3, 7, 3, 3, 3];

  for (let i = 0; i < startDays.length; i++) {
    const allocation = {
      ID: `Alloc${i + 1}`,
      ResourceID: `Res${i + 1}`,
      BarText: `A${i + 1}`,
      my_WorkingDays: workingDays[i],
      Entries: [
        {
          Start: new Date(2028, 0, startDays[i]),
          End: new Date(2028, 0, startDays[i] + workingDays[i]),
          my_EndDay: new Date(2028, 0, startDays[i] + workingDays[i] - 1),
        },
      ],
    };
    allocations.push(allocation);
  }

  return allocations;
}

function enableOrDisableSorting() {
  const btn = document.getElementById("sorting");

  if (btn.value === "Activate table column sorting") {
    btn.value = "Deactivate table column sorting";
    vsWidget.option("interactiveSwitchingOfSortOrderEnabled", true);
  } else {
    btn.value = "Activate table column sorting";
    vsWidget.option({
      interactiveSwitchingOfSortOrderEnabled: false,
      resourceRowSortMode: RowSortMode.None,
    });
  }
}

});
</script>
</body>
</html>

```