

24-01 Resources view - Tooltips

This example shows how allocation data of resources can be displayed in a tooltip.

Essentially, the tooltip window's frame is provided without any predefined content. The appearance and disappearance of the tooltip are managed by the visual scheduling widget.

The tooltip content is best defined using an HTML `<table>` element. As is generally recommended, the styling should be separated from the content structure to maintain a clean and modular design.

A style is defined for the container, the header, the descriptor column field and the data column field:

```
.table-container {
  width: 100%;
  max-width: 400px;
  border-collapse: collapse;
}

.tooltip-header {
  padding: 2px 6px;
  background-color: #ccc;
  color: #444;
  font-size: 13px;
  font-weight: bold;
}

.table-rowHeader {
  padding: 2px 6px;
  background-color: #f1f1f1;
  text-align: left;
  border-bottom: 1px solid #ddd;
  font-size: 12px;
  line-height: 1.2;
}

.table-data {
  padding: 2px 6px;
  text-align: left;
  border-bottom: 1px solid #ddd;
  font-size: 12px;
  line-height: 1.2;
}
```

The structure of the table is as follows:

```
<table class="table-container">
  <tr>
    <td class="tooltip-header" colspan="2">Activity</td>
  </tr>
  <tr>
    <td class="table-rowHeader">ID</td>
    <td class="table-data">{{ID}}</td>
  </tr>
  <tr>
    <td class="table-rowHeader">Name</td>
    <td class="table-data">{{TableText}}</td>
  </tr>
  <tr>
    <td class="table-rowHeader">Start</td>
    <td class="table-data">{{Entries[0].Start:date:de}}</td>
  </tr>
  <tr>
    <td class="table-rowHeader">End</td>
    <td class="table-data">{{Entries[0].End:date:de}}</td>
  </tr>
</table>
```

```

</tr>
<tr>
  <td class="table-rowHeader">Working days</td>
  <td class="table-data">{{WorkingDays}}</td>
</tr>
</table>

```

The double-curly entry contains the data field whose content is to be output in the template.

```
{{my_WorkingDays}}
```

If the data field contains a date, the identifier for a formatting rule can be entered using a colon:

```
{{Entries[0].Start:date:dateTimeFmt}}
```

The format is defined with the **intlDateTimeFormatOptionsMap** option.

```

intlDateTimeFormatOptionsMap: {
  dateTimeFmt: {
    year: "numeric",
    month: "2-digit",
    day: "2-digit",
    hour: "2-digit",
    minute: "2-digit",
  },
},

```

Finally, the widget must know which tooltip template is to be used. To do this, the tooltip templates are added to the **vsWidget** using the **addTooltipTemplates** method

```
vsWidget.addTooltipTemplates(tooltips);
```

and the **defaultAllocationBarTooltipTemplateID** option is set with the corresponding **ID**.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>02-06 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>
</style>
  .table-container {
    width: 100%;
    max-width: 400px;
    border-collapse: collapse;
  }

  .tooltip-header {
    padding: 2px 6px;
    background-color: #ccc;
    color: #444;
    font-size: 13px;
    font-weight: bold;
  }

  .table-rowHeader {
    padding: 2px 6px;
    background-color: #f1f1f1;
    text-align: left;
  }

```

```

        border-bottom: 1px solid #ddd;
        font-size: 12px;
        line-height: 1.2;
    }

    .table-data {
        padding: 2px 6px;
        text-align: left;
        border-bottom: 1px solid #ddd;
        font-size: 12px;
        line-height: 1.2;
    }
}
</style>
</head>
<body>
    <div id="toolbar">
        <h3>
            <a href="docs/24-01_ResourcesView-Tooltips-Tooltips.pdf"
                target="_blank">24-01 Resources view -Tooltips</a>
        </h3>
    </div>
    <div id="VSWidget"></div>
    <script>
        Miscellaneous.ready(() => {
            const { BarDragModes, RowDesigns, SnapTargets, ViewType } = netronic.nVSW;

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

            let options = {
                licenseKey: window.VSW_LICENSE_KEY,

                viewType: ViewType.Resources,

                start: timeAreaStart,
                end: timeAreaEnd,

                intlDateTimeFormatOptionsMap: {
                    dateTimeFmt: {
                        year: "numeric",
                        month: "2-digit",
                        day: "2-digit",
                        hour: "2-digit",
                        minute: "2-digit",
                    },
                },
                ignoreCalendarOnAllocationBarInteractions: true,
                defaultAllocationSnapTargetsForStart: SnapTargets.None,
                defaultAllocationSnapTargetsForEnd: SnapTargets.None,

                defaultResourceExpandedRowDesign: RowDesigns.Bars,
                defaultResourceCollapsedRowDesign: RowDesigns.Bars |
                    RowDesigns.BarsInHiddenDescendantRows |
                    RowDesigns.BarsStacked,

                defaultAllocationAllowedBarDragModes: BarDragModes.DragHorizontally,

                defaultAllocationBarSelectable: false,
                defaultResourceRowSelectable: false,
                multipleSelectionEnabled: false,
                fixedTableColumnWidth: 0,

                defaultAllocationBarTooltipTemplateID: "Tooltip1",

                onDrop: (args) => {
                    const allocation = args.object;
                    allocation.Start = args.newStart;
                    allocation.End = args.newEnd;
                    allocation.Entries[0].Start = args.newStart;
                    allocation.Entries[0].End = args.newEnd;
                    vsWidget.updateAllocations([allocation]);
                    vsWidget.render();
                },
            };
        });
    </script>

```

```

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

const tooltipTemplates = createTooltipsTemplates();
const resources = createResources();
const allocations = createAllocations();
vsWidget.addTooltipTemplates(tooltipTemplates);
vsWidget.addResources(resources);
vsWidget.addAllocations(allocations);
vsWidget.render();

function createResources() {
    const resources = [];

    for (let i = 1; i <= 11; i++) {
        const resource = { ID: `Res${i}` };
        if (i !== 6)
            resource.TableText = `Resource ${i}`;
        if (i > 7)
            resource.ParentID = `Res${i - 1}`;
        resources.push(resource);
    }

    return resources;
}

function createAllocations() {
    const allocations = [];
    const startDays = [5, 6, 15, 20, 25, 0, 5, 6, 15, 20, 25];
    const workingDays = [3, 3, 3, 3, 3, 0, 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}`,
            Name: `Allocation ${i + 1}`,
            my_WorkingDays: workingDays[i],
            Entries: [
                {
                    Start: new Date(2028, 0, startDays[i]),
                    End: new Date(2028, 0, startDays[i] + workingDays[i]),
                },
            ],
        };
        allocations.push(allocation);
    }

    for (let i = 0; i < 5; i++) {
        allocations[i].my_DocUrl = "pdf/02-06_ResourcesView_Tooltips.pdf";
        allocations[i].my_DocName = `Allocation ${i + 1} Document`;
    }

    return allocations;
}

function createTooltipsTemplates() {
    return [
        {
            ID: "Tooltip1",
            IsInteractive: true,
            HTMLFormat: `
                <div style="max-height: 120px; overflow:auto;"
                <table class="table-container">
                    <tr>
                        <td class="tooltip-header" colspan="2">Allocation</td>
                    </tr>
                    <tr>
                        <td class="table-rowHeader">ID</td>
                        <td class="table-data">${ID}</td>
                    </tr>
                    <tr>
                        <td class="table-rowHeader">Name</td>
            `
        }
    ]
}

```

```

        <td class="table-data">{{TableText}}</td>
    </tr>
    <tr>
        <td class="table-rowHeader">Start</td>
        <td class="table-data">{{Entries[0].Start:date:dateTimeFmt}}</td>
    </tr>
    <tr>
        <td class="table-rowHeader">End</td>
        <td class="table-data">{{Entries[0].End:date:dateTimeFmt}}</td>
    </tr>
    <tr>
        <td class="table-rowHeader">Working days</td>
        <td class="table-data">{{my_WorkingDays}}</td>
    </tr>
    <tr>
        <td class="table-rowHeader">Document</td>
        <td class="table-data"><a href="{{my_DocUrl}}"
            target="_blank">{{my_DocName}}</a></td>
    </tr>
</table>
</div>`,
    ],
    },
    });
</script>
</body>
</html>

```