

# js/script\_radarboard\_einzelprojekt\_udim.js

0

## Introduction

This document will walk you through the implementation of [js/script\\_radarboard\\_einzelprojekt\\_udim.js](#) ✓.

The script is responsible for setting up and managing a radar chart visualization. It includes the initialization of the chart, handling themes, creating containers, setting titles, managing events, and updating data dynamically.

We will cover:

1. Initializing the chart and setting themes.
2. Creating containers and setting up the radar chart.
3. Managing titles and menu interactions.
4. Handling data updates and dynamic changes.
5. Managing series and events for the radar chart.

## Code flow

### Initializing the chart and setting themes

We start by initializing the root element for the chart and setting the themes. This is crucial for the chart's appearance and behavior.

 js/script\_radarboard\_einzelprojekt\_udim.js ✓ Up to date

```
1  //#region ///-----A Set root, Loading Indicator, Themes, Container, Title, Filter, Buttons----
2  var root = am5.Root.new("chartdiv");
3
4  //#region ///-----A.2 Themes, Container Chart und Text, Title-----
5  root.setThemes([
6    am5themes_Animated.new(root),
7    am5themes_Responsive.new(root),
8  ]);
```

### Creating containers and setting up the radar chart

Next, we create a container for the radar chart and define its dimensions and layout.

 js/script\_radarboard\_einzelprojekt\_udim.js ✓ Up to date

```
10 ///-----Container und Radar Chart-----
11 var container = root.container.children.push(
12   am5.Container.new(root, {
13     width: myChartWidthEinzel,
14     height: myChartHeightEinzel,
15     layout: root.verticalLayout
16   })
17 );
```

We then set up the radar chart within this container, specifying its scale and layout properties.

 js/script\_radarboard\_einzelfprojekt\_udim.js  Up to date

```
259 //-----ende container & textRadarboard Links-----
260
261 //endregion ///-----A.2 End Themes, Container, Title-----
262
263 //region ///-----A.3 Chart-----
264
265 var chartTechUDimEinzel = container.children.push(am5radar.RadarChart.new(root, {
266     //scale: scaleChartUDim, ///0.92 intiale size chart etwas reduzieren
267     scale: 0.90, ///0.92 intiale size chart etwas reduzieren
268     panX: false,
269     panY: false,
270     //x: am5.percent(10),
271     //dy: 230, //chart nach rechts schieben damit links platz für text box ist
272     y: am5.percent(0),
273     dy: 38,
274     centerX: am5.percent(-11),
275     //startAngle: -125,
276     //endAngle: 235,
277     startAngle: -120,
278     endAngle: 240,
```

## Managing titles and menu interactions

We define the title container and set up the title label, which will be dynamically updated later.

 js/script\_radarboard\_einzelfprojekt\_udim.js  Up to date

```
30 //-----title-----
31 var containerTitle = container.children.push(
32     am5.Container.new(root, {
33         layout: root.horizontalLayout
34     })
35 );
36
37 var title = containerTitle.children.push(am5.Label.new(root, {
38     text: "", //wird später über update Title gesetzt
39     fontSize: 18,
40     fontWeight: 700,
41     paddingLeft: 620,
42     // x: 450,
43 }));
```

We also add a menu label with a tooltip and click event to toggle the visibility of a menu.

 js/script\_radarboard\_einzelfprojekt\_udim.js  Up to date

```

46 //-----Ende pseudo Legend -----
47 //-----Menu Links Hamburger rechts Hilfe Video-----
48 var labelMenu = containerTitle.children.push(am5.Label.new(root, {
49     paddingTop: 7,
50     x: 17,
51     text: "[fontWeight: 500 fontSize: 18px #000]≡[fontWeight: 500 fontSize: 13px #000 ] Menü Visualisierung",
52     fontSize: 18,
53     fill: ColorBlackYAxisText,
54     cursorOverStyle: "pointer",
55     tooltip: am5.Tooltip.new(root, { paddingBottom: 5, paddingTop: 3, pointerOrientation:"left" }),
56     tooltipX: 28,
57     tooltipY: 18,
58     tooltipText: "[fontWeight: 500 fontSize: 12px #fff]Menü zur Auswahl\neiner Visualisierung",
59     background: am5.Rectangle.new(root, {
60         fill: am5.color(0x000000),
61         fillOpacity: 0
62     })
63 }));

```

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date

```

65 labelMenu.events.on("click", function (ev) {
66     if (labelMenu.get("active")) { //Menu war active -> hide div
67         //console.Log("Menu active");
68         labelMenu.set("active", false);
69         document.getElementById('menu-radar').style.display = 'none';
70     } else { //Menu war inactive -> div anzeigen
71         //console.Log("Menu inactive");
72         labelMenu.set("active", true);
73         document.getElementById('menu-radar').style.display = 'inline-block';
74     };
75 });

```

## Handling data updates and dynamic changes

The `updateTitleText`  function dynamically updates the title based on the visibility of the projects.

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date

```

111 function updateTitleText() {
112     var counterVisible = 0; var indexSingle = 0;
113     for (var udim = 1; udim <= projekteNamen.length; udim++) { //array aller projekte durchgehen
114         if (seriesEinzelProjekt[udim].isVisible()) {
115             containerTextRadarboardTech.show(0);
116             counterVisible = counterVisible + 1;
117             indexSingle = udim;
118             textProjektTitel2 = "[#535353]" + projekteNamen[udim - 1]; //array projekteNamen geht ab 0 Los

```

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date

```

134   textProjektTitel = textProjektTitel2 + textProjektTitel3 + textProjektTitel1;
135   title.set("text", textProjektTitel);
136
137  });

```

Similarly, the `updateProjektBeschreibungText` ✓ function updates the project description text based on the visible projects.

js/script\_radarboard\_einzelprojekt\_udim.js ✓ Up to date

```

155  function updateProjektBeschreibungText() {
156    var counterVisible = 0;
157    var indexVisible = -1;
158
159    for (var indexProjekt = 1; indexProjekt <= projekteNamen.length; indexProjekt++) { //array aller projekte durchlaufen
160      if (seriesEinzelProjekt[indexProjekt].isVisible()) {
161        indexVisible = indexProjekt - 1;
162      }
163    };
164    if (indexVisible >= 0) {
165      if (dataUnterDimensionenEinzelTexte[indexVisible]["verbundpartner"] == "") {labelProjektVerbundpartner.hide();}
166      else { labelProjektVerbundpartner.show();}

```

js/script\_radarboard\_einzelprojekt\_udim.js ✓ Up to date

```

176  // textProjektBeschreibung = text1Projekt + "\n[fontSize: 20px][fontSize: 13px]" + text2Projekt + "\n[fontSize: 20px][fontSize: 13px]" + text3Projekt + "\n[fontSize: 20px][fontSize: 13px]";
177  textProjektBeschreibung = text1Projekt + "\n[fontSize: 20px][fontSize: 13px]" + text2Projekt + "\n[fontSize: 20px][fontSize: 13px]" + text3Projekt + "\n[fontSize: 20px][fontSize: 13px]";
178  textRadarboardTech.set("text", textProjektBeschreibung);

```

## Managing series and events for the radar chart

We create series for each project and set up their properties, including colors and data fields.

js/script\_radarboard\_einzelprojekt\_udim.js ✓ Up to date

```

673  seriesEinzelProjekt[index] = chartTechUDimEinzel.series.push(am5radar.RadarLineSeries.new(root, {
674    name: projekteNamen[index - 1], //projekteNamen Array läuft von 0..33 deshalb index-1
675    xAxis: xAxisTechUDimEinzel,
676    yAxis: yAxisTechUDimEinzel,
677    valueYField: "value" + index,
678    categoryXField: "dimension",
679    stroke: color,
680    fill: color,
681
682  }));

```

We also handle the visibility events for these series to update the title and project descriptions accordingly.

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date

```
813 function erstelleLegendEventsTechUDimEinzel(index) {  
814     seriesEinzelProjekt[index].on("visible", function (visible, target) {  
815         updateTitleText();  
816         updateProjektBeschreibungText();  
817  
818     if (visible) {  
819         //console.log("visible: ", index)
```

Finally, we initialize the series data and set up the legend to include all series.

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date


```
907 //chartTechUDimEinzel.set("scale", 0.95); /// intial chart herunterskalieren da evtl. schriften zu gross  
908  
909 ///series mit neuem datensatz initialisieren  
910 for (var index = 1; index <= projekteNamen.length; index++) {  
911     seriesEinzelProjekt[index].data.setAll(dataUnterDimensionenEinzel);  
912 };
```

 js/script\_radarboard\_einzelprojekt\_udim.js  Up to date

```
924 xAxisTechUDimEinzel.data.setAll(dataUnterDimensionenEinzel);  
925  
926 //LegendTechUDimEinzel.data.setAll(chartTechUDimEinzel.series.values);  
927  
928 for (var index = 1; index <= projekteNamen.length; index++) {  
929     legendTechUDimEinzel.data.push(seriesEinzelProjekt[index]);  
930 };
```

## Conclusion

This document has walked you through the implementation of the radar chart visualization in

[js/script\\_radarboard\\_einzelprojekt\\_udim.js](#) . We covered the initialization of the chart, setting themes, creating containers, managing titles and menu interactions, handling data updates, and managing series and events. This setup ensures that the radar chart is dynamically updated and interactive, providing a comprehensive visualization of the projects.