exness

Viz Extension Fundamentals

Pavel Semenov

26 September 2024

Agenda

What is Viz Extension?

Recommended Prerequisites

Anatomy of a Viz Extension

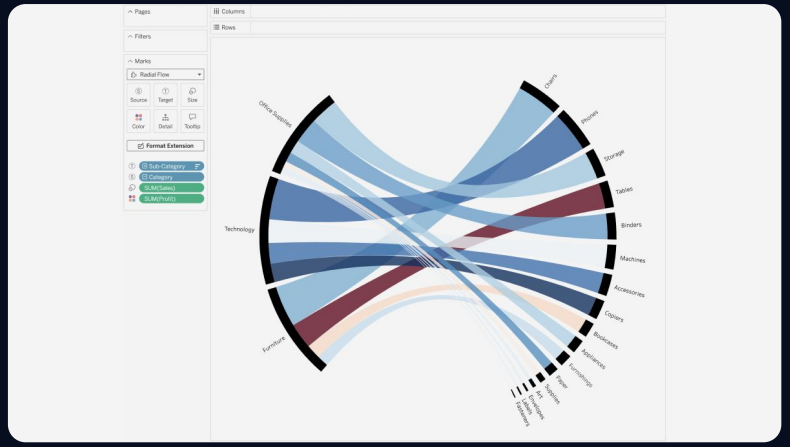
Mission & Data

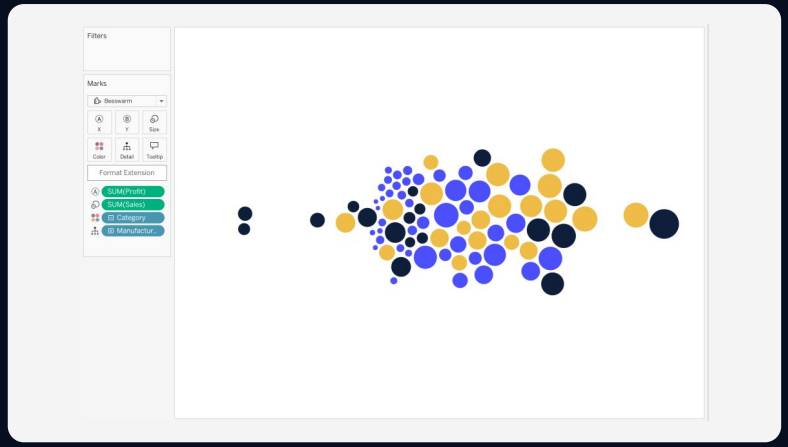
Step-by-step developing process

Q&A

What is Viz Extension?

Viz Extensions are web applications that can extend the native visual capabilities of Tableau.
Viz Extensions give users the ability to interact with custom viz types on their worksheets.





Recommended Prerequisites

- Familiarity with Javascript
- Familiarity with D3 or another JS library that can be used for charting, or ability to pick it up using docs and examples
- Basic understanding of SVG (namely Path elements)
- Ability to follow prompts for command-line scripting
- Experience creating visualizations in Tableau using the Mark Cards

Anatomy of a Viz Extension

.trex

an XML file that describes the extension and provides information to register the extension with Tableau

.html

the web page for your extension that links the Extensions API JavaScript library and other JavaScript, CSS, or HTML resources your extension requires

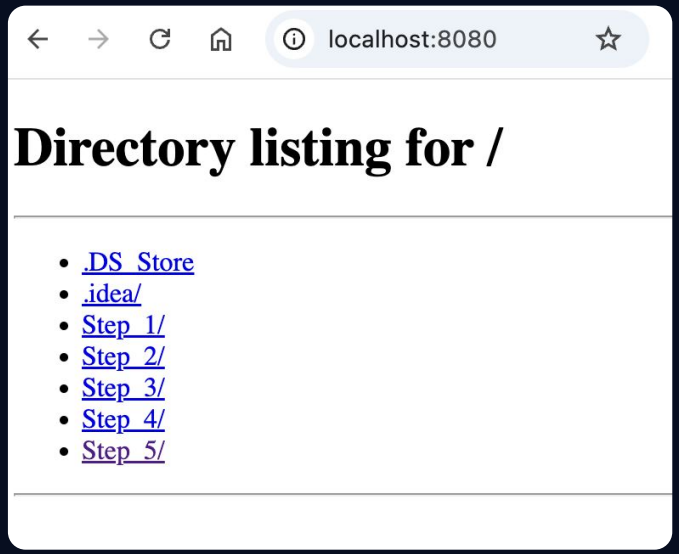
.js

where you initialize your extension and call Extensions API functions, can use JS libraries such as D3, React, Svelte

Preparation

1. Run a local web server for storing our viz extension.

```
python -m http.server 8080 -d "/path/to/extension/content/directory"
```



Preparation

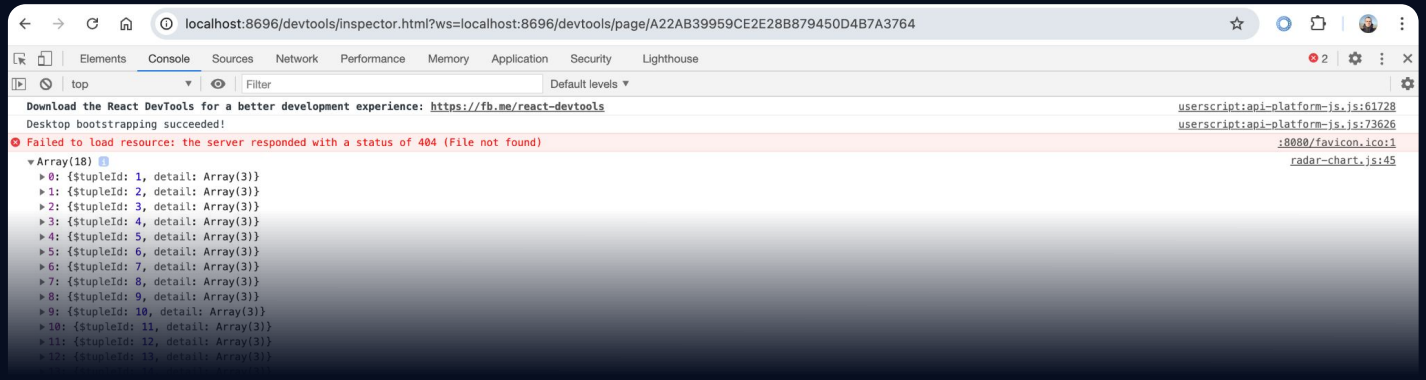
2. Open Tableau Desktop in debug mode

[in Mac Terminal]

```
open "/Applications/Tableau Desktop 2024.2.app" --args --remote-debugging-port=8696
```

[in Windows Command Prompt]

```
"C:\Program Files\Tableau\Tableau 2024.2\bin\tableau.exe" --remote-debugging-port=8696
```

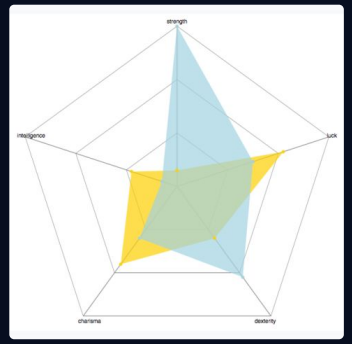


Mission & Data

Mission:

Create a custom Spider (Radar) chart.

Explore company employees performance by their performance metrics using the new chart type.



Data:

	A	B	C
1	**employee**	**metric**	**value**
2	Sarah	ethics	0.7
3	Sarah	problem_solving	0.6
4	Sarah	punctuality	0.3
5	Sarah	job_skills	0.8
6	Sarah	communication	0.9
7	Sarah	team_player	1
8	John	ethics	0.4
9	John	problem_solving	0.2
10	John	punctuality	0.8
11	John	job_skills	0.6
12	John	communication	0.8
13	John	team_player	0.8
14	Kevin	ethics	0.1
15	Kevin	problem_solving	0.5
16	Kevin	punctuality	0.2
17	Kevin	job_skills	0.9
18	Kevin	communication	0.1
19	Kevin	team_player	0.5

Step 1: Empty extension

radar-chart.trex

```
<?xml version="1.0" encoding="utf-8"?>
<manifest manifest-version="0.1" xmlns="http://www.tableau.com/xml/extension_manifest">
  <worksheet-extension id="com.tableau.extension.radar-chart" extension-version="1.0.0">
    <default-locale>en_US</default-locale>
    <name>Radar Chart v1</name>
    <description>Radar Chart v1.0</description>
    <author name="Pavel Semenov" email="pavel.semenov@exness.com" organization="Exness" website="https://www.exness.com"/>
    <min-api-version>1.11</min-api-version>
    <source-location>
      <url>http://localhost:8080/Step_1/radar-chart.html</url>
    </source-location>
    <icon/>
  </worksheet-extension>
</manifest>
```

Step 1: Empty extension

radar-chart.trex

```
<?xml version="1.0" encoding="utf-8"?>
<manifest manifest-version="0.1" xmlns="http://www.tableau.com/xml/extension_manifest">
  <worksheet-extension id="com.tableau.extension.radar-chart" extension-version="1.0.0">
    <default-locale>en_US</default-locale>
    <name>Radar Chart v1</name>
    <description>Radar Chart v1.0</description>
    <author name="Pavel Semenov" email="pavel.semenov@exness.com" organization="Exness" website="https://www.exness.com"/>
    <min-api-version>1.11</min-api-version>
    <source-location>
      <url>http://localhost:8080/Step_1/radar-chart.html</url>
    </source-location>
    <icon/>
  </worksheet-extension>
</manifest>
```

Step 1: Empty extension

radar-chart.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="icon" href="data:,">
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Radar Chart</title>
  <script type="module" src="./tableau.extensions.1.latest.js"></script>
  <link rel="stylesheet" href="./common.css">

</head>
<body>
  <div id="my_dataviz" style="..."></div>
  <script type="module" src="./radar-chart.js"></script>

  <button id="refreshButton" style="...">Refresh</button>
  <script type="module">
    refreshButton.onclick = async () =>
      await import('./radar-chart.js?timestamp=' + performance.now()).then(fetchDataAndRender());
  </script>

</body>
</html>
```

Step 1: Empty extension

radar-chart.js

```
import * as d3 from 'https://cdn.jsdelivr.net/npm/d3@7/+esm';
import {getEncodedData} from './utils.js';

let data : any[] = [];

window.onload = async () : Promise<void> => {
  await tableau.extensions.initializeAsync();
  window.fetchDataAndRender();
  addEventListeners();
};

function getWorksheet() { Show usages
  return tableau.extensions.worksheetContent.worksheet;
}

function getSettings() { Show usages
  return tableau.extensions.settings;
}

window.fetchDataAndRender = async function () : Promise<void> {
  let worksheet = getWorksheet();
  data = await getEncodedData(worksheet);
  await window.renderPlot(data);
};

function addEventListeners() : void { Show usages
  let worksheet = getWorksheet();
  let settings = getSettings();
  worksheet.addEventListener(tableau.TableauEventType.SummaryDataChanged, window.fetchDataAndRender);
  settings.addEventListener(tableau.TableauEventType.SettingsChanged, window.fetchDataAndRender);
  window.onresize = () => window.renderPlot(data, worksheet);
}
```

```
window.renderPlot = async function (data) : Promise<void> {
  const container : HTMLElement = document.getElementById( elementId: 'my_dataviz');
  container.innerHTML = '';

  let width : number = container.clientWidth;
  let height : number = container.clientHeight;

  let svg = d3
      .select(container)
      .append( name: 'svg')
      .attr('width', width)
      .attr('height', height)
      .append( name: 'g')
      .attr('transform', `translate(${width / 2}, ${height / 2})`);

  svg.selectAll("text")
    .data(['Hello Cyprus TUG'])
    .join( separator: "text")
    .style("alignment-baseline", "middle")
    .style("text-anchor", "middle")
    .style("font-size", 40)
    .style("color", "rgb(100, 100, 100)")
    .attr("x", 0)
    .attr("y", 0)
    .text((t) => t);
}
```

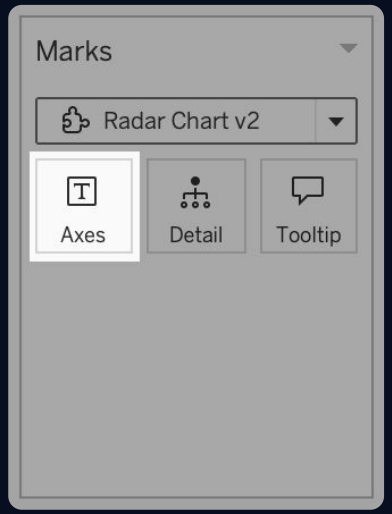
Step 1: Empty extension

Tableau Desktop

Step 2: Spider net

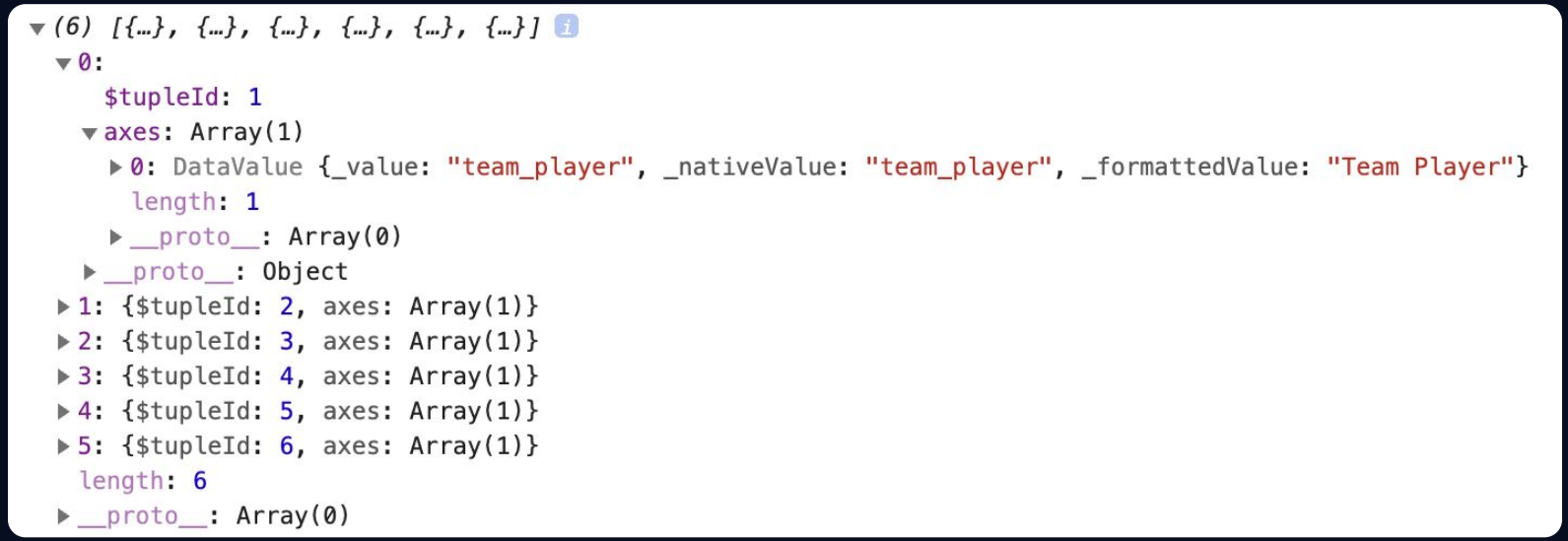
radar-chart.trex

```
<?xml version="1.0" encoding="utf-8"?>
<manifest manifest-version="0.1" xmlns="http://www.tableau.com/xml/extension_manifest">
  <worksheet-extension id="com.tableau.extension.radar-chart" extension-version="1.0.0">
    <default-locale>en_US</default-locale>
    <name>Radar Chart v2</name>
    <description>Radar Chart v1.0</description>
    <author name="Pavel Semenov" email="pavel.semenov@exness.com" organization="Exness" website="https://www.exness.com"/>
    <min-api-version>1.11</min-api-version>
    <source-location>
      <url>http://localhost:8080/Step_2/radar-chart.html</url>
    </source-location>
    <icon/>
    <encoding id="axes">
      <display-name>Axes</display-name>
      <fields max-count="1"/>
      <encoding-icon token="text" />
    </encoding>
  </worksheet-extension>
</manifest>
```



Step 2: Spider net

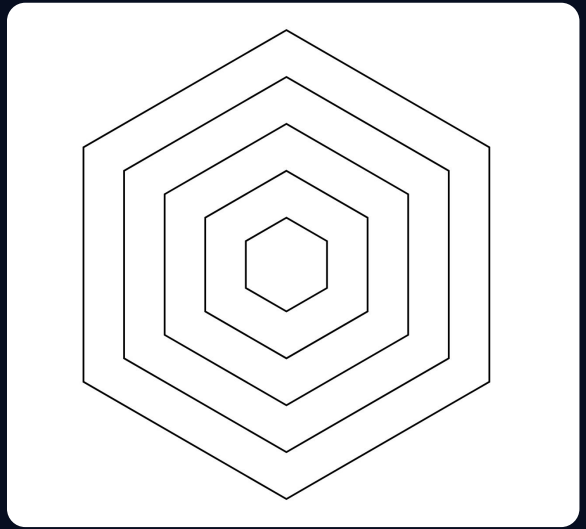
How data looks



Step 2: Spider net

radar-chart.js

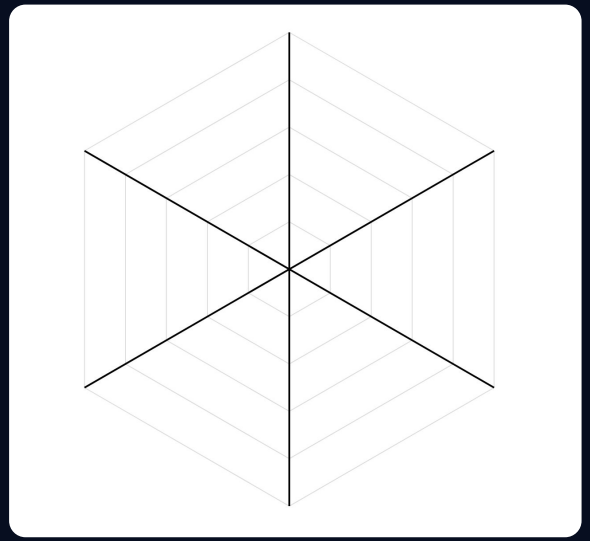
```
let innerRadius : number = 0;
let outerRadius : number = Math.min(width, height) / 2 - 40;

let numberList : number[] = [0.2, 0.4, 0.6, 0.8, 1.0];
let radiusList : unknown[] = numberList.map(r : number => innerRadius + (outerRadius - innerRadius) * r);

const dataLabels = data.map(d => d.axes[0]).reduce((acc, curr) => {
  const exists = acc.some(el => el['value'] === curr['value']);
  return exists ? acc : [...acc, curr];
}, []);

function makePolygonsGrid(r) {  Show usages
  // Code for spider net drawing
  // including angle and length calculation for each axis and radius
}

svg.selectAll('g.grid')
  .data(radiusList)
  .enter().append( name: 'g')
  .each(makePolygonsGrid);
```



Step 2: Spider net

radar-chart.js

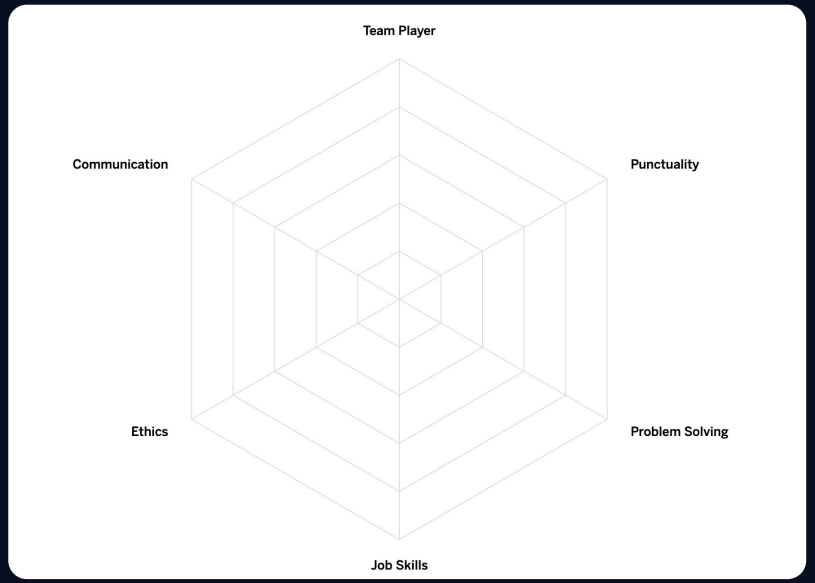
```
function drawLines(data) {  Show usages
  // Axis angle calculating based on the number of axes
  // and then drawing axes lines
}
```



Step 2: Spider net

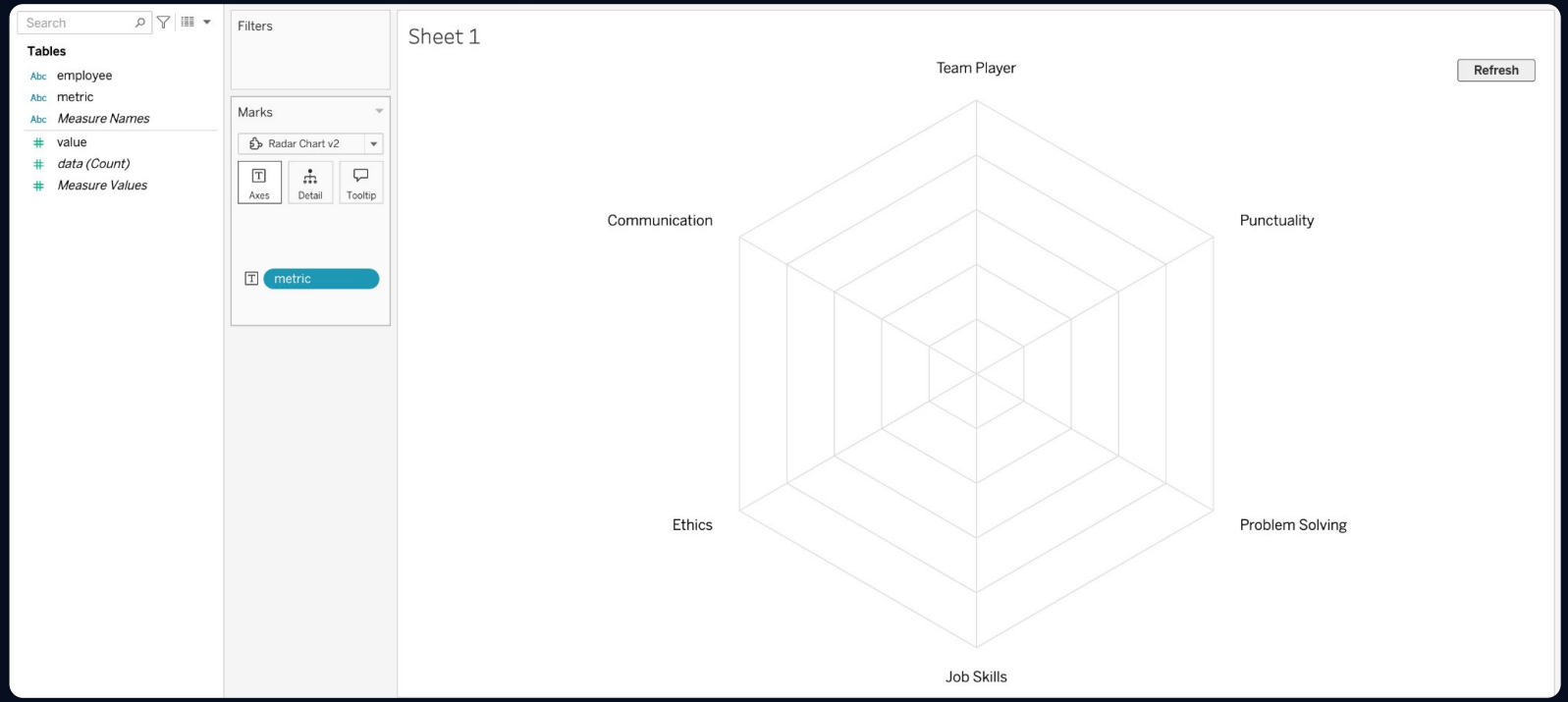
radar-chart.js

```
function drawLabels(data) {  Show usages
  // Axis angle calculating based on the number of axes
  // and then drawing Labels on the end of each axis
}
```



Step 2: Spider net

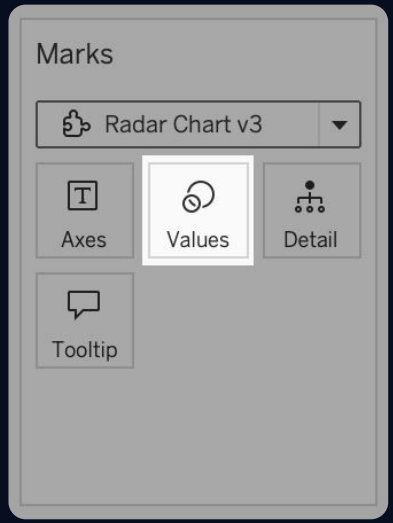
Tableau Desktop



Step 3: Area

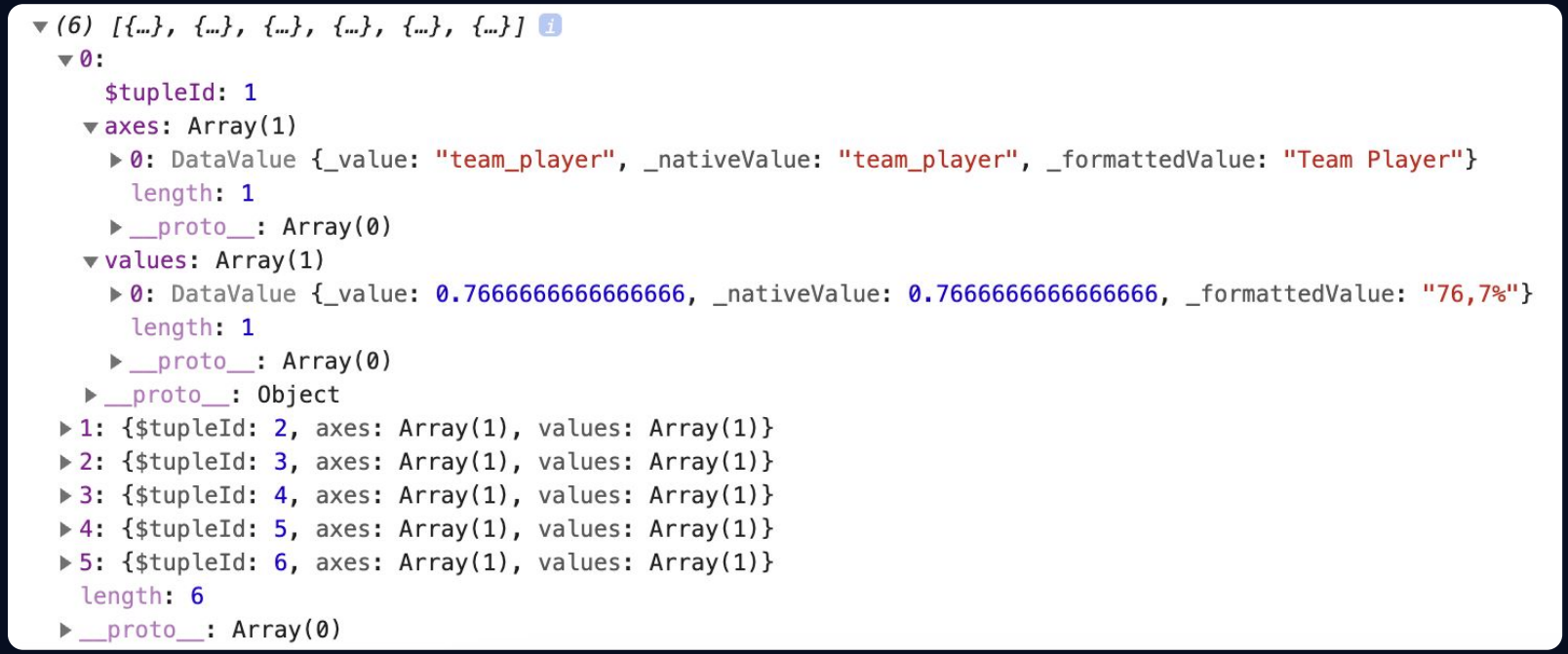
radar-chart.trex

```
<icon/>
<encoding id="axes">
  <display-name>Axes</display-name>
  <fields max-count="1"/>
  <encoding-icon token="text" />
</encoding>
<encoding id="values">
  <display-name>Values</display-name>
  <role-spec>
    <role-type>continuous-measure</role-type>
    <role-type>continuous-dimension</role-type>
  </role-spec>
  <fields max-count="1"/>
  <encoding-icon token="size" />
</encoding>
```



Step 3: Area

How data looks



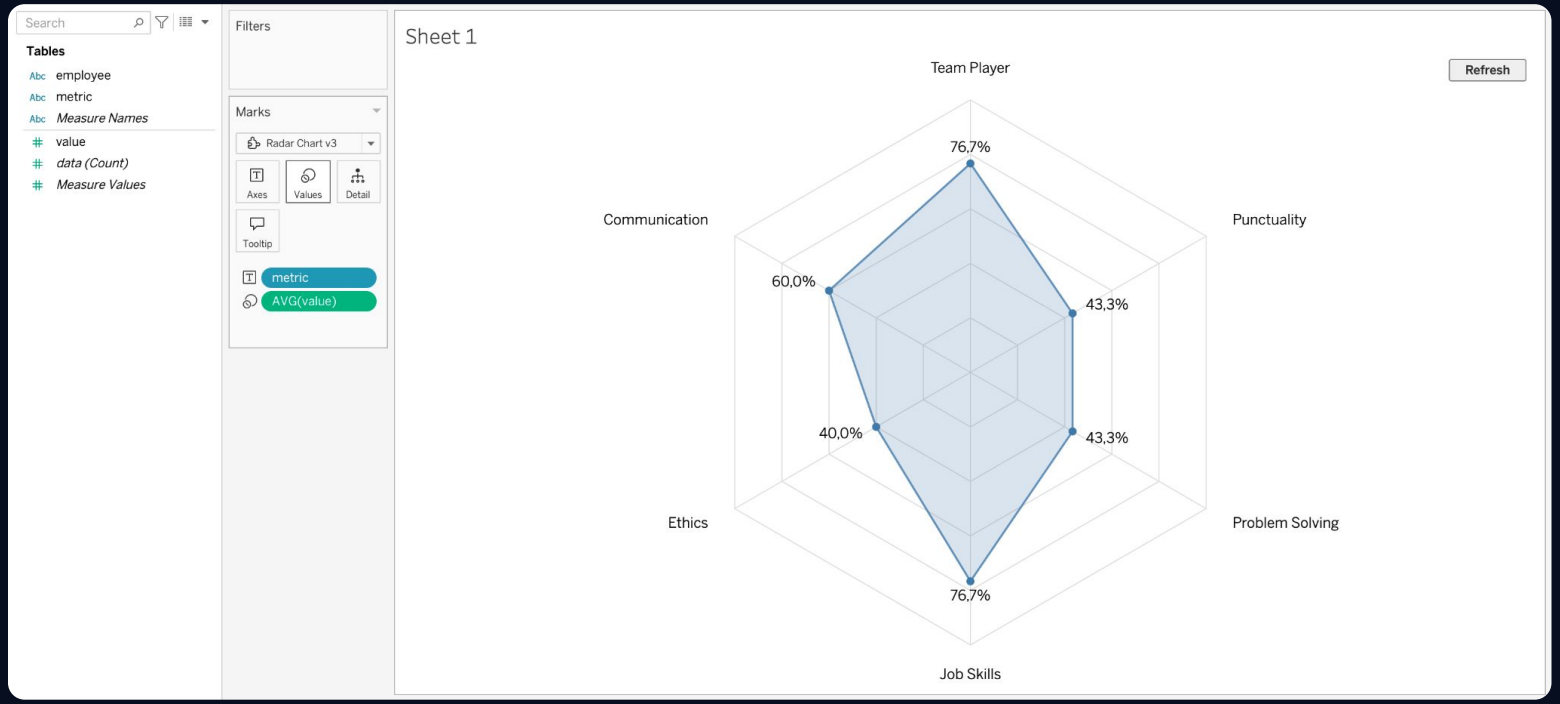
Step 3: Area

radar-chart.js

```
function drawArea(data) {  Show usages
  // Draw polygon based on metrics values
}

function drawAreaPoints(data) {  Show usages
  // Draw clickable vertices points based on metrics values
}

function drawAreaLabels(data) {  Show usages
  // Draw values as a text at the end of each vertex
}
```

Step 3: Area

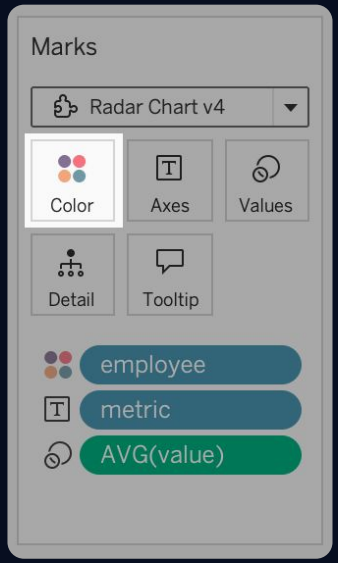
Tableau Desktop



Step 4: Multiple areas

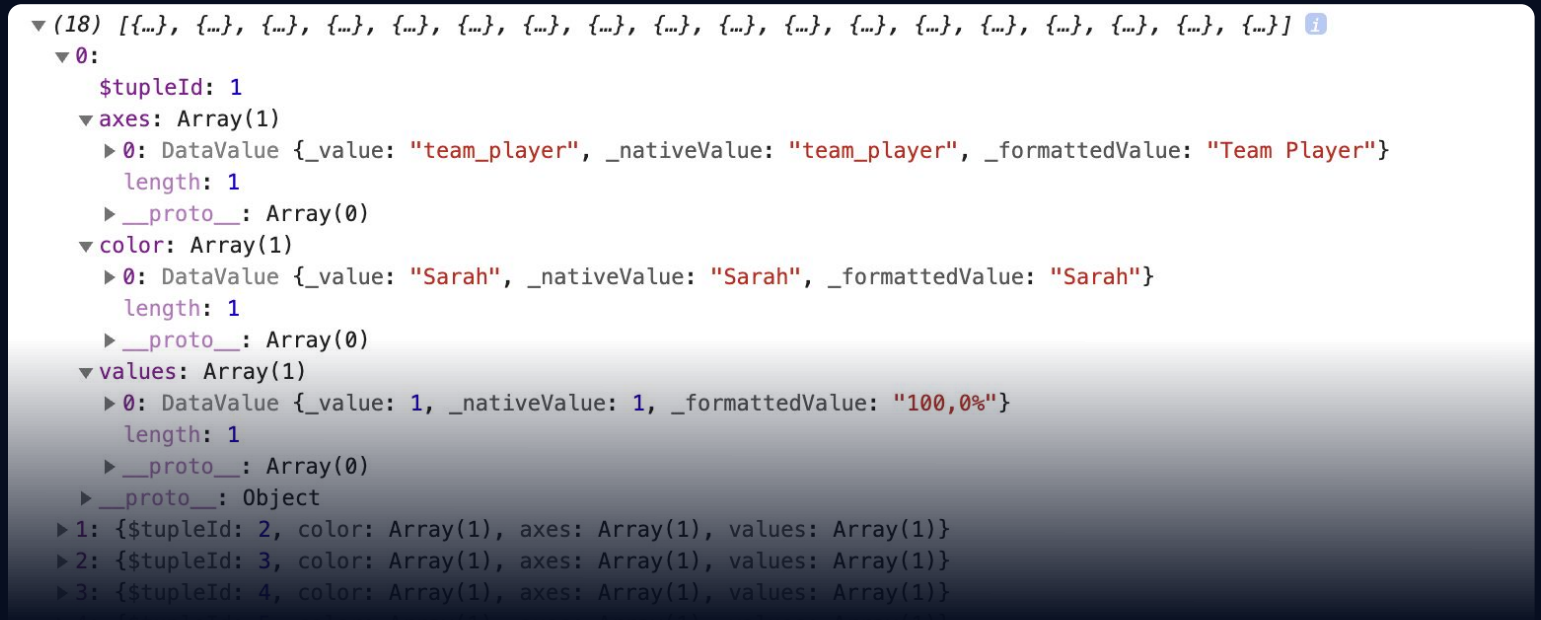
radar-chart.trex

```
<source-location>
  <url>http://localhost:8080/Step_4/radar-chart.html</url>
</source-location>
<icon/>
<encoding id="color">
  <display-name>Color</display-name>
  <fields max-count="1"/>
  <encoding-icon token="color" />
</encoding>
<encoding id="axes">
  <display-name>Axes</display-name>
  <fields max-count="1"/>
  <encoding-icon token="text" />
</encoding>
<encoding id="values">
  <display-name>Values</display-name>
  <role-spec>
    <role-type>continuous-measure</role-type>
    <role-type>continuous-dimension</role-type>
```



Step 4: Multiple areas

How data looks



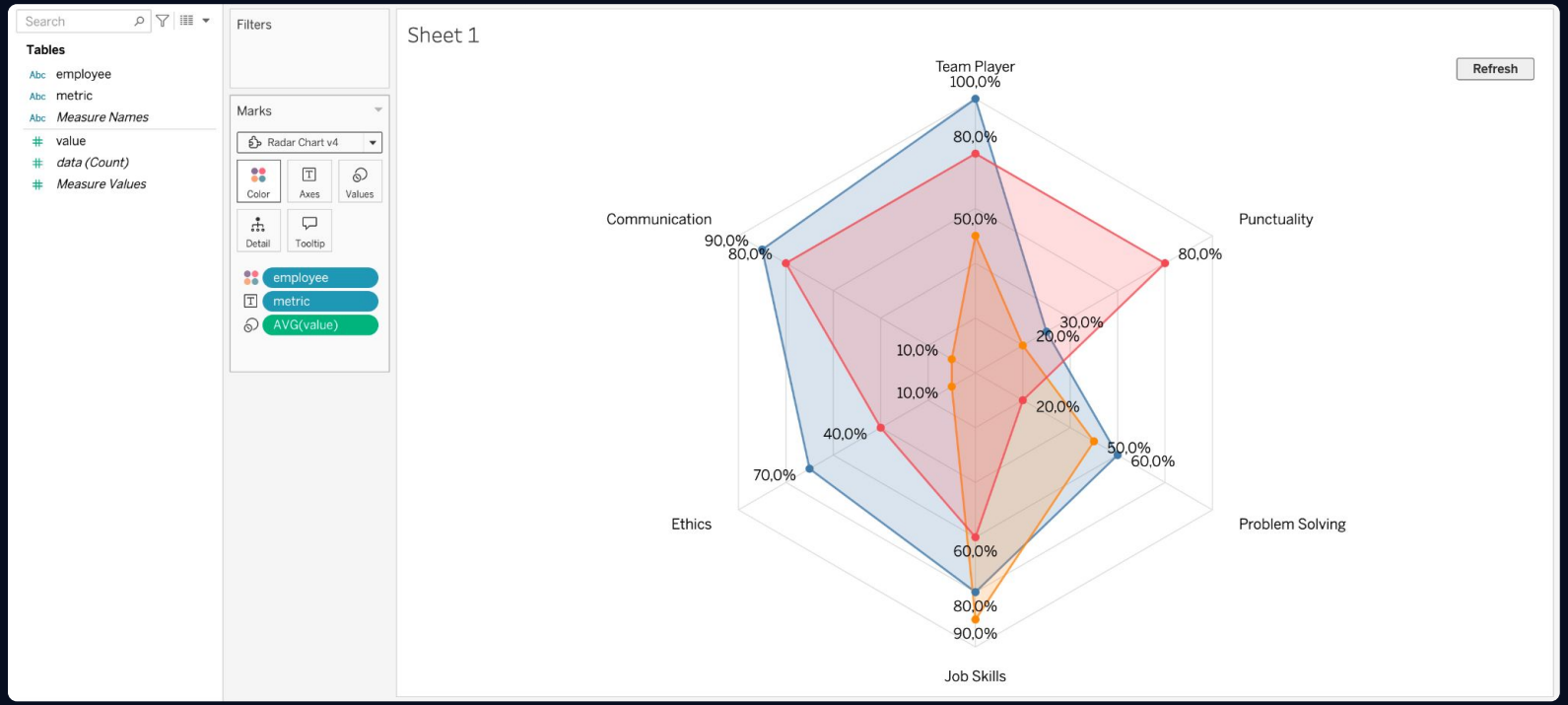
Step 4: Multiple areas

radar-chart.js

```
function drawAreaModified(data) {  Show usages
  // Draw polygon based on metrics values
  // Check if Color Mark is not empty then use it to split the data by Color Mark
  // Actually also by Detail Mark for more native chart
}

function drawAreaPointsModified(data) {  Show usages
  // Draw clickable vertices points based on metrics values
  // Check if Color Mark is not empty then use it to split the data by Color Mark
  // Actually also by Detail Mark for more native chart
}

function drawAreaLabelsModified(data) {  Show usages
  // Draw values as a text at the end of each vertex
  // Check if Color Mark is not empty then use it to split the data by Color Mark
  // Actually also by Detail Mark for more native chart
}
```

Step 4: Multiple areas

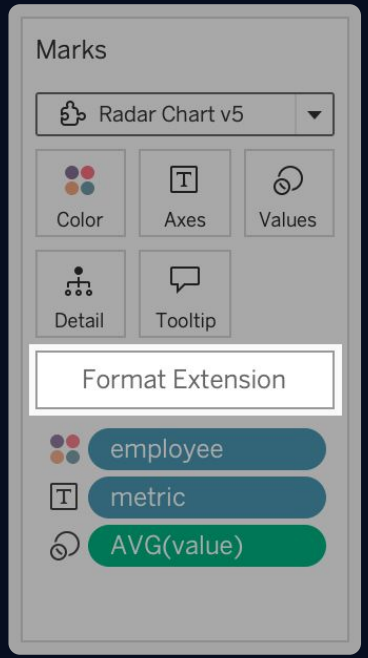
Tableau Desktop



Step 5: Extension settings

radar-chart.trex

```
<source-location>
  <url>http://localhost:8080/Step_5/radar-chart.html</url>
</source-location>
<icon/>
<context-menu>
    <configure-context-menu-item />
</context-menu>
<encoding id="color">
  <display-name>Color</display-name>
  <fields max-count="1"/>
  <encoding-icon token="color" />
</encoding>
<encoding id="axes">
  <display-name>Axes</display-name>
  <fields max-count="1"/>
  <encoding-icon token="text" />
</encoding>
<encoding id="values">
```



Step 5: Extension settings

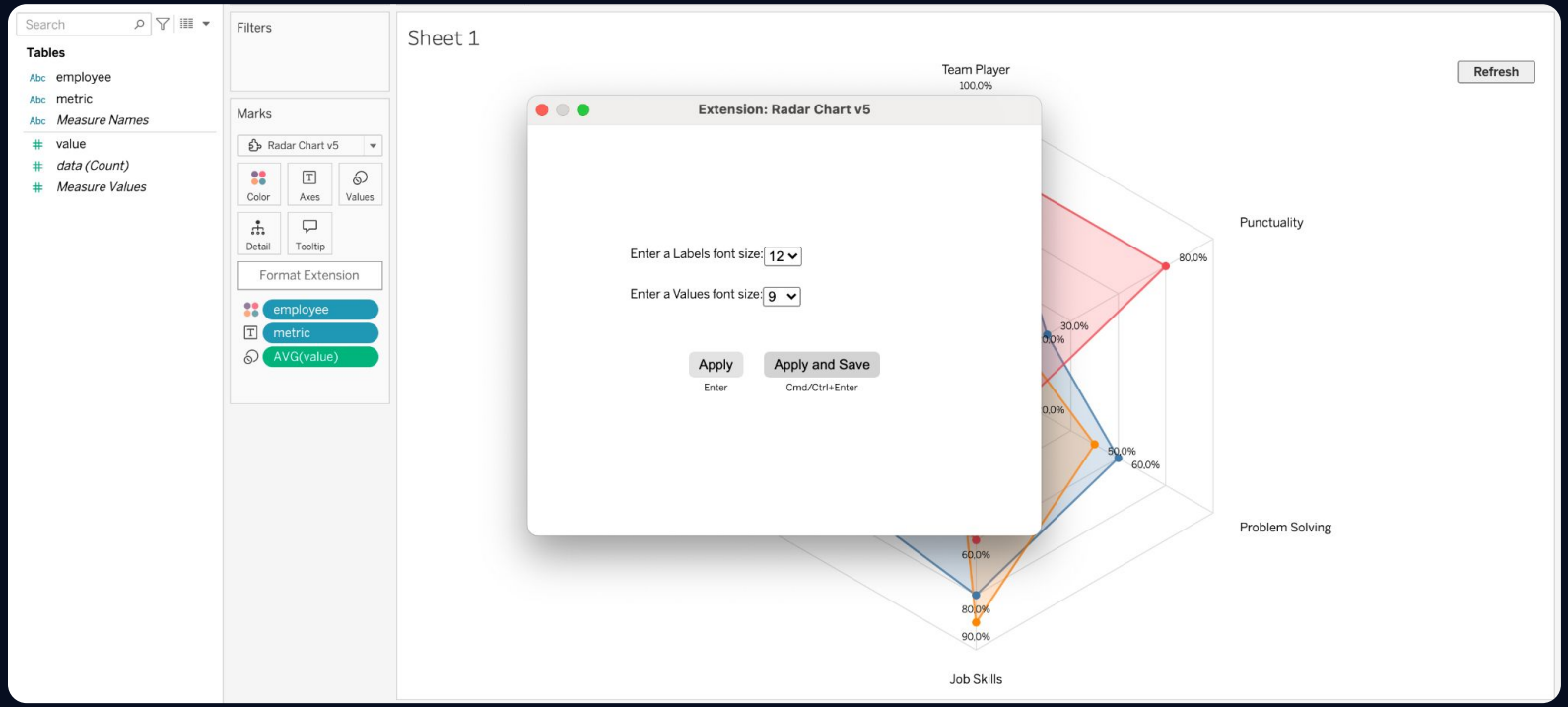
radar-chart.js

```
      console.error(error.message);
    }
  });
}

function configure() {  Show usages
  // Code for opening window with settings
  // such as font sizes, area colors, etc.
  // and updating chart after saving settings
}

window.onload = async () : Promise<void> => {
  await tableau.extensions.initializeAsync({'configure': configure});
  window.fetchDataAndRender();
  addEventListeners();
};

function getWorksheet() {  Show usages
  return tableau.extensions.worksheetContent.worksheet;
```

+ radar-chart-dialog.html
+ radar-chart-dialog.js

Step 5: Extension settings

Tableau Desktop



How to publish it?

- It is not needed to publish it exactly on Tableau Exchange, you can keep using it locally

- You need to set up a web server to provide access to the extension for all your Tableau users
  [http://localhost:8080 → https://your-tableau-extension.com]

- You need to adjust your Tableau Server or Tableau Cloud

- Learn more:
  Tableau Server
  Tableau Cloud

Thank you!

Content

LinkedIn

Telegram