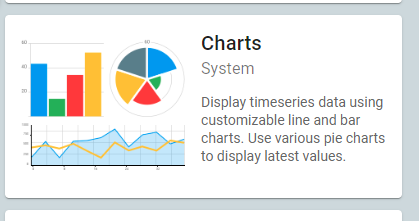
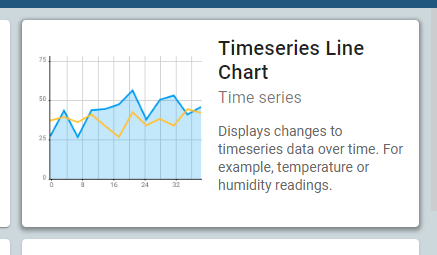
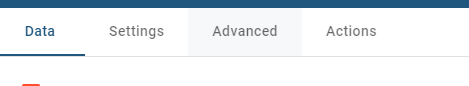
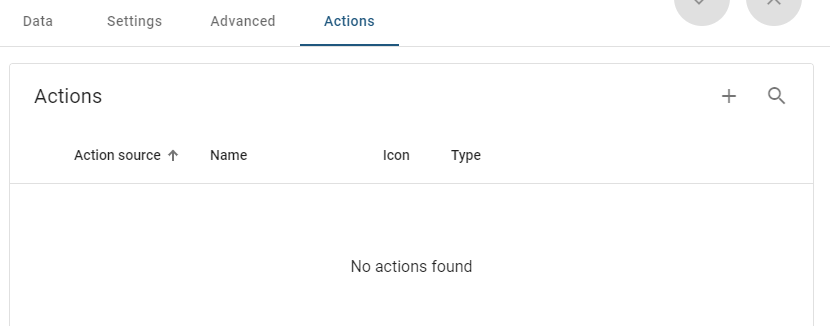
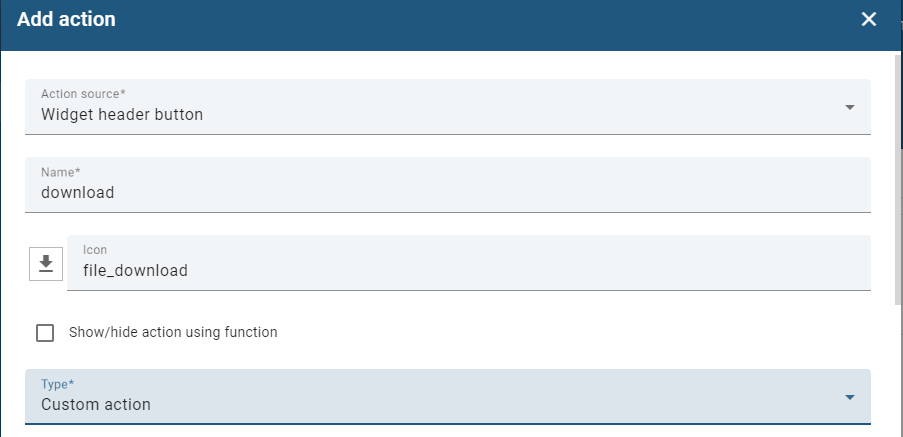
**Dashboard**  
  
Add New Widget ->   
  
Chart ->   
  
  
  
TimeSeries Line Chart ->   
  
  
  
Edit Widget ->  
  
  
  
 Action -> Add Action ->  
  
****  
  
 Type -> Custom Action  
  
  
  
**Following Function:**  
  
function convertToCSV(data) {

const dataObject = {};

const csvRows = [];

const headersKey = Object.keys(data[0].data[0]);

for (const rows of data) {

dataObject[rows.dataKey.label] = [];

for (const row of rows.data) {

dataObject[rows.dataKey.label].push(headersKey

.map(header => {

const escaped = ('' + row[

header]).replace(/"/g,

'\\"');

return "${escaped}";

}));

}

}

let titles = Object.keys(dataObject);

const timeData = dataObject[titles[0]].map(data => data[

0]);

csvRows.push(['time', ...titles].join(','));

for (let index = 0; index < timeData.length; index++) {

let tmp = [timeData[index]];

for (const title of titles) {

tmp.push(dataObject[title][index][1]);

}

csvRows.push(tmp.join(','));

}

return csvRows.join('\n');

}

function downloadCSV(csvContent, fileName) {

const link = document.createElement('a');

link.setAttribute('href',

'data:text/csv;charset=utf-8,' +

encodeURIComponent(csvContent));

link.setAttribute('download', fileName);

link.style.display = 'none';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

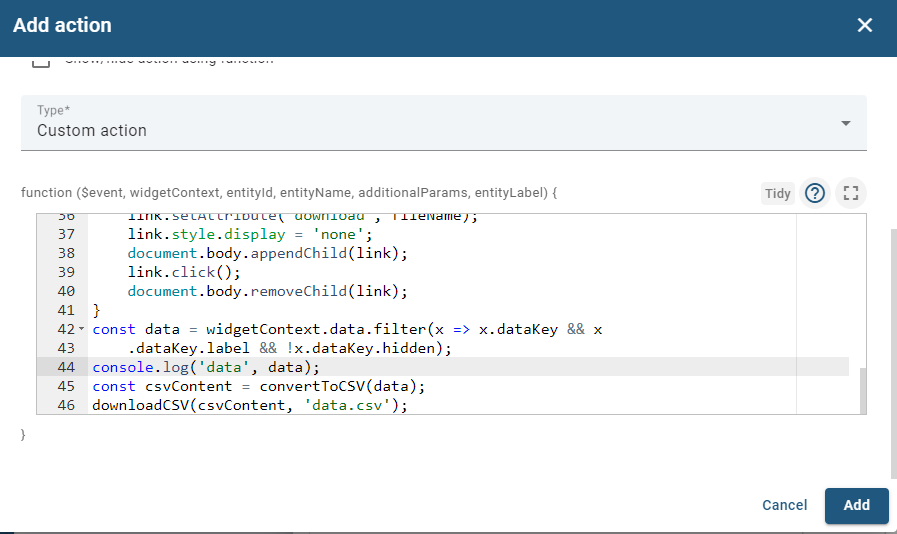
}

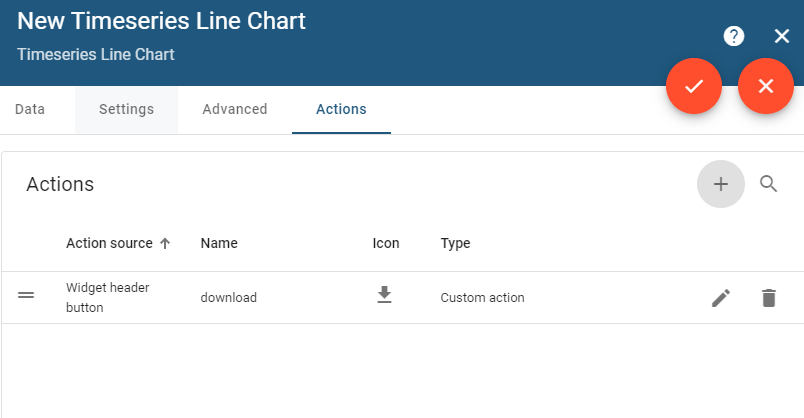
const data = widgetContext.data.filter(x => x.dataKey && x

.dataKey.label && !x.dataKey.hidden);

console.log('data', data);

const csvContent = convertToCSV(data);

downloadCSV(csvContent, 'data.csv');  
  


Add ->  
  
  
  
Apply Changes   
  
Done Time Line Chart will be displayed.