 [Chart.js](http://docs.google.com/docs/3.9.1/)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

* [Information](http://docs.google.com/docs/3.9.1/samples/information.html)
* Bar Charts
* Line Charts
* Other charts
* Area charts
* Scales
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
  + [Data Decimation](http://docs.google.com/docs/3.9.1/samples/advanced/data-decimation.html)
  + [Derived Axis Type](http://docs.google.com/docs/3.9.1/samples/advanced/derived-axis-type.html)
  + [Derived Chart Type](http://docs.google.com/docs/3.9.1/samples/advanced/derived-chart-type.html)
  + [Linear Gradient](http://docs.google.com/docs/3.9.1/samples/advanced/linear-gradient.html)
  + [Programmatic Event Triggers](http://docs.google.com/docs/3.9.1/samples/advanced/programmatic-events.html)
  + [Animation Progress Bar](http://docs.google.com/docs/3.9.1/samples/advanced/progress-bar.html)
  + [Radial Gradient](http://docs.google.com/docs/3.9.1/samples/advanced/radial-gradient.html)
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Programmatic Event Triggers

hover tooltip actions config setup

function triggerHover(chart) { if (chart.getActiveElements().length > 0) { chart.setActiveElements([]); } else { chart.setActiveElements([ { datasetIndex: 0, index: 0, }, { datasetIndex: 1, index: 0, } ]); } chart.update(); }

function triggerHover(chart) {  
 if (chart.getActiveElements().length > 0) {  
 chart.setActiveElements([]);  
 } else {  
 chart.setActiveElements([  
 {  
 datasetIndex: 0,  
 index: 0,  
 }, {  
 datasetIndex: 1,  
 index: 0,  
 }  
 ]);  
 }  
 chart.update();  
}

function triggerTooltip(chart) { const tooltip = chart.tooltip; if (tooltip.getActiveElements().length > 0) { tooltip.setActiveElements([], {x: 0, y: 0}); } else { const chartArea = chart.chartArea; tooltip.setActiveElements([ { datasetIndex: 0, index: 2, }, { datasetIndex: 1, index: 2, } ], { x: (chartArea.left + chartArea.right) / 2, y: (chartArea.top + chartArea.bottom) / 2, }); } chart.update(); }

function triggerTooltip(chart) {  
 const tooltip = chart.tooltip;  
 if (tooltip.getActiveElements().length > 0) {  
 tooltip.setActiveElements([], {x: 0, y: 0});  
 } else {  
 const chartArea = chart.chartArea;  
 tooltip.setActiveElements([  
 {  
 datasetIndex: 0,  
 index: 2,  
 }, {  
 datasetIndex: 1,  
 index: 2,  
 }  
 ],  
 {  
 x: (chartArea.left + chartArea.right) / 2,  
 y: (chartArea.top + chartArea.bottom) / 2,  
 });  
 }  
 chart.update();  
}

const actions = [ { name: 'Trigger Hover', handler: triggerHover }, { name: 'Trigger Tooltip', handler: triggerTooltip } ];

const actions = [  
 {  
 name: 'Trigger Hover',  
 handler: triggerHover  
 },  
 {  
 name: 'Trigger Tooltip',  
 handler: triggerTooltip  
 }  
];

const config = { type: 'bar', data: data, options: { }, };

const config = {  
 type: 'bar',  
 data: data,  
 options: {  
 },  
};

const DATA\_COUNT = 7; const NUMBER\_CFG = {count: DATA\_COUNT, min: -100, max: 100}; const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [ { label: 'Dataset 1', data: Utils.numbers(NUMBER\_CFG), borderColor: Utils.CHART\_COLORS.red, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5), hoverBorderWidth: 5, hoverBorderColor: 'green', }, { label: 'Dataset 2', data: Utils.numbers(NUMBER\_CFG), borderColor: Utils.CHART\_COLORS.blue, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5), hoverBorderWidth: 5, hoverBorderColor: 'green', } ] };

const DATA\_COUNT = 7;  
const NUMBER\_CFG = {count: DATA\_COUNT, min: -100, max: 100};  
const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [  
 {  
 label: 'Dataset 1',  
 data: Utils.numbers(NUMBER\_CFG),  
 borderColor: Utils.CHART\_COLORS.red,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5),  
 hoverBorderWidth: 5,  
 hoverBorderColor: 'green',  
 },  
 {  
 label: 'Dataset 2',  
 data: Utils.numbers(NUMBER\_CFG),  
 borderColor: Utils.CHART\_COLORS.blue,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5),  
 hoverBorderWidth: 5,  
 hoverBorderColor: 'green',  
 }  
 ]  
};

## [**#**](#30j0zll) API

* [Chart](http://docs.google.com/docs/3.9.1/api/classes/Chart.html)
  + [setActiveElements](http://docs.google.com/docs/3.9.1/api/classes/Chart.html#setactiveelements)
* [TooltipModel](http://docs.google.com/docs/3.9.1/api/interfaces/TooltipModel.html)
  + [setActiveElements](http://docs.google.com/docs/3.9.1/api/interfaces/TooltipModel.html#setactiveelements)

## [**#**](#1fob9te) Docs

* [Bar](http://docs.google.com/docs/3.9.1/charts/bar.html)
  + [Interactions (hoverBorderColor)](http://docs.google.com/docs/3.9.1/charts/bar.html#interactions)
* [Interactions](http://docs.google.com/docs/3.9.1/configuration/interactions.html)
* [Tooltip](http://docs.google.com/docs/3.9.1/configuration/tooltip.html)

Last Updated: 8/3/2022, 12:46:38 PM

←  [Linear Gradient](http://docs.google.com/docs/3.9.1/samples/advanced/linear-gradient.html)   [Animation Progress Bar](http://docs.google.com/docs/3.9.1/samples/advanced/progress-bar.html)  →