 [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
  + [Bubble](http://docs.google.com/docs/3.9.1/samples/other-charts/bubble.html)
  + [Combo bar/line](http://docs.google.com/docs/3.9.1/samples/other-charts/combo-bar-line.html)
  + [Doughnut](http://docs.google.com/docs/3.9.1/samples/other-charts/doughnut.html)
  + [Multi Series Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/multi-series-pie.html)
  + [Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/pie.html)
  + [Polar area](http://docs.google.com/docs/3.9.1/samples/other-charts/polar-area.html)
  + [Polar area centered point labels](http://docs.google.com/docs/3.9.1/samples/other-charts/polar-area-center-labels.html)
  + [Radar](http://docs.google.com/docs/3.9.1/samples/other-charts/radar.html)
  + [Radar skip points](http://docs.google.com/docs/3.9.1/samples/other-charts/radar-skip-points.html)
  + [Scatter](http://docs.google.com/docs/3.9.1/samples/other-charts/scatter.html)
  + [Scatter - Multi axis](http://docs.google.com/docs/3.9.1/samples/other-charts/scatter-multi-axis.html)
  + [Stacked bar/line](http://docs.google.com/docs/3.9.1/samples/other-charts/stacked-bar-line.html)
* Area charts
* Scales
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Combo bar/line

config setup actions

const config = { type: 'bar', data: data, options: { responsive: true, plugins: { legend: { position: 'top', }, title: { display: true, text: 'Chart.js Combined Line/Bar Chart' } } }, };

const config = {  
 type: 'bar',  
 data: data,  
 options: {  
 responsive: true,  
 plugins: {  
 legend: {  
 position: 'top',  
 },  
 title: {  
 display: true,  
 text: 'Chart.js Combined Line/Bar Chart'  
 }  
 }  
 },  
};

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), order: 1 }, { label: 'Dataset 2', data: Utils.numbers(NUMBER\_CFG), borderColor: Utils.CHART\_COLORS.blue, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5), type: 'line', order: 0 } ] };

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),  
 order: 1  
 },  
 {  
 label: 'Dataset 2',  
 data: Utils.numbers(NUMBER\_CFG),  
 borderColor: Utils.CHART\_COLORS.blue,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5),  
 type: 'line',  
 order: 0  
 }  
 ]  
};

const actions = [ { name: 'Randomize', handler(chart) { chart.data.datasets.forEach(dataset => { dataset.data = Utils.numbers({count: chart.data.labels.length, min: -100, max: 100}); }); chart.update(); } }, { name: 'Add Dataset', handler(chart) { const data = chart.data; const dsColor = Utils.namedColor(chart.data.datasets.length); const newDataset = { label: 'Dataset ' + (data.datasets.length + 1), backgroundColor: Utils.transparentize(dsColor, 0.5), borderColor: dsColor, borderWidth: 1, data: Utils.numbers({count: data.labels.length, min: -100, max: 100}), }; chart.data.datasets.push(newDataset); chart.update(); } }, { name: 'Add Data', handler(chart) { const data = chart.data; if (data.datasets.length > 0) { data.labels = Utils.months({count: data.labels.length + 1}); for (let index = 0; index < data.datasets.length; ++index) { data.datasets[index].data.push(Utils.rand(-100, 100)); } chart.update(); } } }, { name: 'Remove Dataset', handler(chart) { chart.data.datasets.pop(); chart.update(); } }, { name: 'Remove Data', handler(chart) { chart.data.labels.splice(-1, 1); // remove the label first chart.data.datasets.forEach(dataset => { dataset.data.pop(); }); chart.update(); } } ];

const actions = [  
 {  
 name: 'Randomize',  
 handler(chart) {  
 chart.data.datasets.forEach(dataset => {  
 dataset.data = Utils.numbers({count: chart.data.labels.length, min: -100, max: 100});  
 });  
 chart.update();  
 }  
 },  
 {  
 name: 'Add Dataset',  
 handler(chart) {  
 const data = chart.data;  
 const dsColor = Utils.namedColor(chart.data.datasets.length);  
 const newDataset = {  
 label: 'Dataset ' + (data.datasets.length + 1),  
 backgroundColor: Utils.transparentize(dsColor, 0.5),  
 borderColor: dsColor,  
 borderWidth: 1,  
 data: Utils.numbers({count: data.labels.length, min: -100, max: 100}),  
 };  
 chart.data.datasets.push(newDataset);  
 chart.update();  
 }  
 },  
 {  
 name: 'Add Data',  
 handler(chart) {  
 const data = chart.data;  
 if (data.datasets.length > 0) {  
 data.labels = Utils.months({count: data.labels.length + 1});  
 for (let index = 0; index < data.datasets.length; ++index) {  
 data.datasets[index].data.push(Utils.rand(-100, 100));  
 }  
 chart.update();  
 }  
 }  
 },  
 {  
 name: 'Remove Dataset',  
 handler(chart) {  
 chart.data.datasets.pop();  
 chart.update();  
 }  
 },  
 {  
 name: 'Remove Data',  
 handler(chart) {  
 chart.data.labels.splice(-1, 1); // remove the label first  
 chart.data.datasets.forEach(dataset => {  
 dataset.data.pop();  
 });  
 chart.update();  
 }  
 }  
];

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

* [Bar](http://docs.google.com/docs/3.9.1/charts/bar.html)
* [Line](http://docs.google.com/docs/3.9.1/charts/line.html)
* [Data structures (labels)](http://docs.google.com/docs/3.9.1/general/data-structures.html)

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

←  [Bubble](http://docs.google.com/docs/3.9.1/samples/other-charts/bubble.html)   [Doughnut](http://docs.google.com/docs/3.9.1/samples/other-charts/doughnut.html)  →