 [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
  + [Delay](http://docs.google.com/docs/3.9.1/samples/animations/delay.html)
  + [Drop](http://docs.google.com/docs/3.9.1/samples/animations/drop.html)
  + [Loop](http://docs.google.com/docs/3.9.1/samples/animations/loop.html)
  + [Progressive Line](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line.html)
  + [Progressive Line With Easing](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line-easing.html)
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Delay

config setup actions

let delayed; const config = { type: 'bar', data: data, options: { animation: { onComplete: () => { delayed = true; }, delay: (context) => { let delay = 0; if (context.type === 'data' && context.mode === 'default' && !delayed) { delay = context.dataIndex \* 300 + context.datasetIndex \* 100; } return delay; }, }, scales: { x: { stacked: true, }, y: { stacked: true } } } };

let delayed;  
const config = {  
 type: 'bar',  
 data: data,  
 options: {  
 animation: {  
 onComplete: () => {  
 delayed = true;  
 },  
 delay: (context) => {  
 let delay = 0;  
 if (context.type === 'data' && context.mode === 'default' && !delayed) {  
 delay = context.dataIndex \* 300 + context.datasetIndex \* 100;  
 }  
 return delay;  
 },  
 },  
 scales: {  
 x: {  
 stacked: true,  
 },  
 y: {  
 stacked: true  
 }  
 }  
 }  
};

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), backgroundColor: Utils.CHART\_COLORS.red, }, { label: 'Dataset 2', data: Utils.numbers(NUMBER\_CFG), backgroundColor: Utils.CHART\_COLORS.blue, }, { label: 'Dataset 3', data: Utils.numbers(NUMBER\_CFG), backgroundColor: Utils.CHART\_COLORS.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),  
 backgroundColor: Utils.CHART\_COLORS.red,  
 },  
 {  
 label: 'Dataset 2',  
 data: Utils.numbers(NUMBER\_CFG),  
 backgroundColor: Utils.CHART\_COLORS.blue,  
 },  
 {  
 label: 'Dataset 3',  
 data: Utils.numbers(NUMBER\_CFG),  
 backgroundColor: Utils.CHART\_COLORS.green,  
 },  
 ]  
};

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(); } }, ];

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();  
 }  
 },  
];

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

* [Animations](http://docs.google.com/docs/3.9.1/configuration/animations.html)
  + [animation (delay)](http://docs.google.com/docs/3.9.1/configuration/animations.html#animation)
  + [Animation Callbacks](http://docs.google.com/docs/3.9.1/configuration/animations.html#animation-callbacks)
* [Bar](http://docs.google.com/docs/3.9.1/charts/bar.html)
  + [Stacked Bar Chart](http://docs.google.com/docs/3.9.1/charts/bar.html#stacked-bar-chart)
* [Options](http://docs.google.com/docs/3.9.1/general/options.html)
  + [Scriptable Options](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options)

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

←  [Radar Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/radar.html)   [Drop](http://docs.google.com/docs/3.9.1/samples/animations/drop.html)  →