 [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
  + [Linear Scale - Min-Max](http://docs.google.com/docs/3.9.1/samples/scales/linear-min-max.html)
  + [Linear Scale - Suggested Min-Max](http://docs.google.com/docs/3.9.1/samples/scales/linear-min-max-suggested.html)
  + [Linear Scale - Step Size](http://docs.google.com/docs/3.9.1/samples/scales/linear-step-size.html)
  + [Log Scale](http://docs.google.com/docs/3.9.1/samples/scales/log.html)
  + [Stacked Linear / Category](http://docs.google.com/docs/3.9.1/samples/scales/stacked.html)
  + [Time Scale](http://docs.google.com/docs/3.9.1/samples/scales/time-line.html)
  + [Time Scale - Max Span](http://docs.google.com/docs/3.9.1/samples/scales/time-max-span.html)
  + [Time Scale - Combo Chart](http://docs.google.com/docs/3.9.1/samples/scales/time-combo.html)
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Time Scale - Max Span

config setup actions

const config = { type: 'line', data: data, options: { spanGaps: 1000 \* 60 \* 60 \* 24 \* 2, // 2 days responsive: true, interaction: { mode: 'nearest', }, plugins: { title: { display: true, text: 'Chart.js Time - spanGaps: 172800000 (2 days in ms)' }, }, scales: { x: { type: 'time', display: true, title: { display: true, text: 'Date' }, ticks: { autoSkip: false, maxRotation: 0, major: { enabled: true }, // color: function(context) { // return context.tick && context.tick.major ? '#FF0000' : 'rgba(0,0,0,0.1)'; // }, font: function(context) { if (context.tick && context.tick.major) { return { weight: 'bold', }; } } } }, y: { display: true, title: { display: true, text: 'value' } } } }, };

const config = {  
 type: 'line',  
 data: data,  
 options: {  
 spanGaps: 1000 \* 60 \* 60 \* 24 \* 2, // 2 days  
 responsive: true,  
 interaction: {  
 mode: 'nearest',  
 },  
 plugins: {  
 title: {  
 display: true,  
 text: 'Chart.js Time - spanGaps: 172800000 (2 days in ms)'  
 },  
 },  
 scales: {  
 x: {  
 type: 'time',  
 display: true,  
 title: {  
 display: true,  
 text: 'Date'  
 },  
 ticks: {  
 autoSkip: false,  
 maxRotation: 0,  
 major: {  
 enabled: true  
 },  
 // color: function(context) {  
 // return context.tick && context.tick.major ? '#FF0000' : 'rgba(0,0,0,0.1)';  
 // },  
 font: function(context) {  
 if (context.tick && context.tick.major) {  
 return {  
 weight: 'bold',  
 };  
 }  
 }  
 }  
 },  
 y: {  
 display: true,  
 title: {  
 display: true,  
 text: 'value'  
 }  
 }  
 }  
 },  
};

const data = { datasets: [{ label: 'Dataset with string point data', backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5), borderColor: Utils.CHART\_COLORS.red, fill: false, data: [{ x: Utils.newDateString(0), y: Utils.rand(0, 100) }, { x: Utils.newDateString(2), y: Utils.rand(0, 100) }, { x: Utils.newDateString(4), y: Utils.rand(0, 100) }, { x: Utils.newDateString(6), y: Utils.rand(0, 100) }], }, { label: 'Dataset with date object point data', backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5), borderColor: Utils.CHART\_COLORS.blue, fill: false, data: [{ x: Utils.newDate(0), y: Utils.rand(0, 100) }, { x: Utils.newDate(2), y: Utils.rand(0, 100) }, { x: Utils.newDate(5), y: Utils.rand(0, 100) }, { x: Utils.newDate(6), y: Utils.rand(0, 100) }] }] };

const data = {  
 datasets: [{  
 label: 'Dataset with string point data',  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5),  
 borderColor: Utils.CHART\_COLORS.red,  
 fill: false,  
 data: [{  
 x: Utils.newDateString(0),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDateString(2),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDateString(4),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDateString(6),  
 y: Utils.rand(0, 100)  
 }],  
 }, {  
 label: 'Dataset with date object point data',  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5),  
 borderColor: Utils.CHART\_COLORS.blue,  
 fill: false,  
 data: [{  
 x: Utils.newDate(0),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDate(2),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDate(5),  
 y: Utils.rand(0, 100)  
 }, {  
 x: Utils.newDate(6),  
 y: Utils.rand(0, 100)  
 }]  
 }]  
};

const actions = [ { name: 'Randomize', handler(chart) { chart.data.datasets.forEach(dataset => { dataset.data.forEach(function(dataObj, j) { const newVal = Utils.rand(0, 100); if (typeof dataObj === 'object') { dataObj.y = newVal; } else { dataset.data[j] = newVal; } }); }); chart.update(); } }, ];

const actions = [  
 {  
 name: 'Randomize',  
 handler(chart) {  
 chart.data.datasets.forEach(dataset => {  
 dataset.data.forEach(function(dataObj, j) {  
 const newVal = Utils.rand(0, 100);  
 if (typeof dataObj === 'object') {  
 dataObj.y = newVal;  
 } else {  
 dataset.data[j] = newVal;  
 }  
 });  
 });  
 chart.update();  
 }  
 },  
];

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

* [Line](http://docs.google.com/docs/3.9.1/charts/line.html)
  + [spanGaps](http://docs.google.com/docs/3.9.1/charts/line.html#line-styling)
* [Time Scale](http://docs.google.com/docs/3.9.1/axes/cartesian/time.html)

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

←  [Time Scale](http://docs.google.com/docs/3.9.1/samples/scales/time-line.html)   [Time Scale - Combo Chart](http://docs.google.com/docs/3.9.1/samples/scales/time-combo.html)  →