 [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)

* [Chart.js](http://docs.google.com/docs/3.9.1/)
* Getting Started
  + [Getting Started](http://docs.google.com/docs/3.9.1/getting-started/)
  + [Installation](http://docs.google.com/docs/3.9.1/getting-started/installation.html)
  + [Integration](http://docs.google.com/docs/3.9.1/getting-started/integration.html)
  + [Usage](http://docs.google.com/docs/3.9.1/getting-started/usage.html)
  + [3.x Migration Guide](http://docs.google.com/docs/3.9.1/getting-started/v3-migration.html)
* General
* Configuration
* Chart Types
* Axes
* Developers

[**#**](#gjdgxs) 3.x Migration Guide

Chart.js 3.0 introduces a number of breaking changes. Chart.js 2.0 was released in April 2016. In the years since then, as Chart.js has grown in popularity and feature set, we've learned some lessons about how to better create a charting library. In order to improve performance, offer new features, and improve maintainability, it was necessary to break backwards compatibility, but we aimed to do so only when worth the benefit. Some major highlights of v3 include:

* Large [performance](http://docs.google.com/docs/3.9.1/general/performance.html) improvements including the ability to skip data parsing and render charts in parallel via webworkers
* Additional configurability and scriptable options with better defaults
* Completely rewritten animation system
* Rewritten filler plugin with numerous bug fixes
* Documentation migrated from GitBook to Vuepress
* API documentation generated and verified by TypeDoc
* No more CSS injection
* Tons of bug fixes
* Tree shaking

## [**#**](#30j0zll) End user migration

### [**#**](#1fob9te) Setup and installation

* Distributed files are now in lower case. For example: dist/chart.js.
* Chart.js is no longer providing the Chart.bundle.js and Chart.bundle.min.js. Please see the [installation](http://docs.google.com/docs/3.9.1/getting-started/installation.html) and [integration](http://docs.google.com/docs/3.9.1/getting-started/integration.html) docs for details on the recommended way to setup Chart.js if you were using these builds.
* moment is no longer specified as an npm dependency. If you are using the time or timeseries scales, you must include one of [the available adapters (opens new window)](https://github.com/chartjs/awesome#adapters) and corresponding date library. You no longer need to exclude moment from your build.
* The Chart constructor will throw an error if the canvas/context provided is already in use
* Chart.js 3 is tree-shakeable. So if you are using it as an npm module in a project and want to make use of this feature, you need to import and register the controllers, elements, scales and plugins you want to use, for a list of all the available items to import see [integration](http://docs.google.com/docs/3.9.1/getting-started/integration.html#bundlers-webpack-rollup-etc). You will not have to call register if importing Chart.js via a script tag or from the [auto](http://docs.google.com/docs/3.9.1/getting-started/integration.html#bundlers-webpack-rollup-etc) register path as an npm module, in this case you will not get the tree shaking benefits. Here is an example of registering components:

import { Chart, LineController, LineElement, PointElement, LinearScale, Title } from `chart.js`  
Chart.register(LineController, LineElement, PointElement, LinearScale, Title);  
const chart = new Chart(ctx, {  
 type: 'line',  
 // data: ...  
 options: {  
 plugins: {  
 title: {  
 display: true,  
 text: 'Chart Title'  
 }  
 },  
 scales: {  
 x: {  
 type: 'linear'  
 },  
 y: {  
 type: 'linear'  
 }  
 }  
 }  
})

### [**#**](#3znysh7) Chart types

* horizontalBar chart type was removed. Horizontal bar charts can be configured using the new [indexAxis](http://docs.google.com/docs/3.9.1/charts/bar.html#horizontal-bar-chart) option

### [**#**](#2et92p0) Options

A number of changes were made to the configuration options passed to the Chart constructor. Those changes are documented below.

#### [**#**](#tyjcwt) Generic changes

* Indexable options are now looping. backgroundColor: ['red', 'green'] will result in alternating 'red' / 'green' if there are more than 2 data points.
* The input properties of object data can now be freely specified, see [data structures](http://docs.google.com/docs/3.9.1/general/data-structures.html) for details.
* Most options are resolved utilizing proxies, instead of merging with defaults. In addition to easily enabling different resolution routes for different contexts, it allows using other resolved options in scriptable options.
  + Options are by default scriptable and indexable, unless disabled for some reason.
  + Scriptable options receive a option resolver as second parameter for accessing other options in same context.
  + Resolution falls to upper scopes, if no match is found earlier. See [options](http://docs.google.com/docs/3.9.1/general/options.html) for details.

#### [**#**](#3dy6vkm) Specific changes

* elements.rectangle is now elements.bar
* hover.animationDuration is now configured in animation.active.duration
* responsiveAnimationDuration is now configured in animation.resize.duration
* Polar area elements.arc.angle is now configured in degrees instead of radians.
* Polar area startAngle option is now consistent with Radar, 0 is at top and value is in degrees. Default is changed from -½π to 0.
* Doughnut rotation option is now in degrees and 0 is at top. Default is changed from -½π to 0.
* Doughnut circumference option is now in degrees. Default is changed from 2π to 360.
* Doughnut cutoutPercentage was renamed to cutoutand accepts pixels as number and percent as string ending with %.
* scale option was removed in favor of options.scales.r (or any other scale id, with axis: 'r')
* scales.[x/y]Axes arrays were removed. Scales are now configured directly to options.scales object with the object key being the scale Id.
* scales.[x/y]Axes.barPercentage was moved to dataset option barPercentage
* scales.[x/y]Axes.barThickness was moved to dataset option barThickness
* scales.[x/y]Axes.categoryPercentage was moved to dataset option categoryPercentage
* scales.[x/y]Axes.maxBarThickness was moved to dataset option maxBarThickness
* scales.[x/y]Axes.minBarLength was moved to dataset option minBarLength
* scales.[x/y]Axes.scaleLabel was renamed to scales[id].title
* scales.[x/y]Axes.scaleLabel.labelString was renamed to scales[id].title.text
* scales.[x/y]Axes.ticks.beginAtZero was renamed to scales[id].beginAtZero
* scales.[x/y]Axes.ticks.max was renamed to scales[id].max
* scales.[x/y]Axes.ticks.min was renamed to scales[id].min
* scales.[x/y]Axes.ticks.reverse was renamed to scales[id].reverse
* scales.[x/y]Axes.ticks.suggestedMax was renamed to scales[id].suggestedMax
* scales.[x/y]Axes.ticks.suggestedMin was renamed to scales[id].suggestedMin
* scales.[x/y]Axes.ticks.unitStepSize was removed. Use scales[id].ticks.stepSize
* scales.[x/y]Axes.ticks.userCallback was renamed to scales[id].ticks.callback
* scales.[x/y]Axes.time.format was renamed to scales[id].time.parser
* scales.[x/y]Axes.time.max was renamed to scales[id].max
* scales.[x/y]Axes.time.min was renamed to scales[id].min
* scales.[x/y]Axes.zeroLine\* options of axes were removed. Use scriptable scale options instead.
* The dataset option steppedLine was removed. Use stepped
* The chart option showLines was renamed to showLine to match the dataset option.
* The chart option startAngle was moved to radial scale options.
* To override the platform class used in a chart instance, pass platform: PlatformClass in the config object. Note that the class should be passed, not an instance of the class.
* aspectRatio defaults to 1 for doughnut, pie, polarArea, and radar charts
* TimeScale does not read t from object data by default anymore. The default property is x or y, depending on the orientation. See [data structures](http://docs.google.com/docs/3.9.1/general/data-structures.html) for details on how to change the default.
* tooltips namespace was renamed to tooltip to match the plugin name
* legend, title and tooltip namespaces were moved from options to options.plugins.
* tooltips.custom was renamed to plugins.tooltip.external

#### [**#**](#1t3h5sf) Defaults

* global namespace was removed from defaults. So Chart.defaults.global is now Chart.defaults
* Dataset controller defaults were relocate to overrides. For example Chart.defaults.line is now Chart.overrides.line
* default prefix was removed from defaults. For example Chart.defaults.global.defaultColor is now Chart.defaults.color
* defaultColor was split to color, borderColor and backgroundColor
* defaultFontColor was renamed to color
* defaultFontFamily was renamed to font.family
* defaultFontSize was renamed to font.size
* defaultFontStyle was renamed to font.style
* defaultLineHeight was renamed to font.lineHeight
* Horizontal Bar default tooltip mode was changed from 'index' to 'nearest' to match vertical bar charts
* legend, title and tooltip namespaces were moved from Chart.defaults to Chart.defaults.plugins.
* elements.line.fill default changed from true to false.
* Line charts no longer override the default interaction mode. Default is changed from 'index' to 'nearest'.

#### [**#**](#4d34og8) Scales

The configuration options for scales is the largest change in v3. The xAxes and yAxes arrays were removed and axis options are individual scales now keyed by scale ID.

The v2 configuration below is shown with it's new v3 configuration

options: {  
 scales: {  
 xAxes: [{  
 id: 'x',  
 type: 'time',  
 display: true,  
 title: {  
 display: true,  
 text: 'Date'  
 },  
 ticks: {  
 major: {  
 enabled: true  
 },  
 font: function(context) {  
 if (context.tick && context.tick.major) {  
 return {  
 weight: 'bold',  
 color: '#FF0000'  
 };  
 }  
 }  
 }  
 }],  
 yAxes: [{  
 id: 'y',  
 display: true,  
 title: {  
 display: true,  
 text: 'value'  
 }  
 }]  
 }  
}

And now, in v3:

options: {  
 scales: {  
 x: {  
 type: 'time',  
 display: true,  
 title: {  
 display: true,  
 text: 'Date'  
 },  
 ticks: {  
 major: {  
 enabled: true  
 },  
 color: (context) => context.tick && context.tick.major && '#FF0000',  
 font: function(context) {  
 if (context.tick && context.tick.major) {  
 return {  
 weight: 'bold'  
 };  
 }  
 }  
 }  
 },  
 y: {  
 display: true,  
 title: {  
 display: true,  
 text: 'value'  
 }  
 }  
 }  
}

* The time scale option distribution: 'series' was removed and a new scale type timeseries was introduced in its place
* In the time scale, autoSkip is now enabled by default for consistency with the other scales

#### [**#**](#2s8eyo1) Animations

Animation system was completely rewritten in Chart.js v3. Each property can now be animated separately. Please see [animations](http://docs.google.com/docs/3.9.1/configuration/animations.html) docs for details.

#### [**#**](#17dp8vu) Customizability

* custom attribute of elements was removed. Please use scriptable options
* The hover property of scriptable options context object was renamed to active to align it with the datalabels plugin.

#### [**#**](#3rdcrjn) Interactions

* To allow DRY configuration, a root options scope for common interaction options was added. options.hover and options.plugins.tooltip now both extend from options.interaction. Defaults are defined at defaults.interaction level, so by default hover and tooltip interactions share the same mode etc.
* interactions are now limited to the chart area + allowed overflow
* {mode: 'label'} was replaced with {mode: 'index'}
* {mode: 'single'} was replaced with {mode: 'nearest', intersect: true}
* modes['X-axis'] was replaced with {mode: 'index', intersect: false}
* options.onClick is now limited to the chart area
* options.onClick and options.onHover now receive the chart instance as a 3rd argument
* options.onHover now receives a wrapped event as the first parameter. The previous first parameter value is accessible via event.native.
* options.hover.onHover was removed, use options.onHover.

#### [**#**](#26in1rg) Ticks

* options.gridLines was renamed to options.grid
* options.gridLines.offsetGridLines was renamed to options.grid.offset.
* options.gridLines.tickMarkLength was renamed to options.grid.tickLength.
* options.ticks.fixedStepSize is no longer used. Use options.ticks.stepSize.
* options.ticks.major and options.ticks.minor were replaced with scriptable options for tick fonts.
* Chart.Ticks.formatters.linear was renamed to Chart.Ticks.formatters.numeric.
* options.ticks.backdropPaddingX and options.ticks.backdropPaddingY were replaced with options.ticks.backdropPadding in the radial linear scale.

#### [**#**](#lnxbz9) Tooltip

* xLabel and yLabel were removed. Please use label and formattedValue
* The filter option will now be passed additional parameters when called and should have the method signature function(tooltipItem, index, tooltipItems, data)
* The custom callback now takes a context object that has tooltip and chart properties
* All properties of tooltip model related to the tooltip options have been moved to reside within the options property.
* The callbacks no longer are given a data parameter. The tooltip item parameter contains the chart and dataset instead
* The tooltip item's index parameter was renamed to dataIndex and value was renamed to formattedValue
* The xPadding and yPadding options were merged into a single padding object

## [**#**](#35nkun2) Developer migration

While the end-user migration for Chart.js 3 is fairly straight-forward, the developer migration can be more complicated. Please reach out for help in the #dev [Slack (opens new window)](https://chartjs-slack.herokuapp.com/) channel if tips on migrating would be helpful.

Some of the biggest things that have changed:

* There is a completely rewritten and more performant animation system.
  + Element.\_model and Element.\_view are no longer used and properties are now set directly on the elements. You will have to use the method getProps to access these properties inside most methods such as inXRange/inYRange and getCenterPoint. Please take a look at [the Chart.js-provided elements (opens new window)](https://github.com/chartjs/Chart.js/tree/master/src/elements) for examples.
  + When building the elements in a controller, it's now suggested to call updateElement to provide the element properties. There are also methods such as getSharedOptions and includeOptions that have been added to skip redundant computation. Please take a look at [the Chart.js-provided controllers (opens new window)](https://github.com/chartjs/Chart.js/tree/master/src/controllers) for examples.
* Scales introduced a new parsing API. This API takes user data and converts it into a more standard format. E.g. it allows users to provide numeric data as a string and converts it to a number where necessary. Previously this was done on the fly as charts were rendered. Now it's done up front with the ability to skip it for better performance if users provide data in the correct format. If you're using standard data format like x/y you may not need to do anything. If you're using a custom data format you will have to override some of the parse methods in core.datasetController.js. An example can be found in [chartjs-chart-financial (opens new window)](https://github.com/chartjs/chartjs-chart-financial), which uses an {o, h, l, c} data format.

A few changes were made to controllers that are more straight-forward, but will affect all controllers:

* Options:
  + global was removed from the defaults namespace as it was unnecessary and sometimes inconsistent
  + Dataset defaults are now under the chart type options instead of vice-versa. This was not able to be done when introduced in 2.x for backwards compatibility. Fixing it removes the biggest stumbling block that new chart developers encountered
  + Scale default options need to be updated as described in the end user migration section (e.g. x instead of xAxes and y instead of yAxes)
* updateElement was changed to updateElements and has a new method signature as described below. This provides performance enhancements such as allowing easier reuse of computations that are common to all elements and reducing the number of function calls

### [**#**](#1ksv4uv) Removed

The following properties and methods were removed:

#### [**#**](#44sinio) Removed from Chart

* Chart.animationService
* Chart.active
* Chart.borderWidth
* Chart.chart.chart
* Chart.Bar. New charts are created via new Chart and providing the appropriate type parameter
* Chart.Bubble. New charts are created via new Chart and providing the appropriate type parameter
* Chart.Chart
* Chart.Controller
* Chart.Doughnut. New charts are created via new Chart and providing the appropriate type parameter
* Chart.innerRadius now lives on doughnut, pie, and polarArea controllers
* Chart.lastActive
* Chart.Legend was moved to Chart.plugins.legend.\_element and made private
* Chart.Line. New charts are created via new Chart and providing the appropriate type parameter
* Chart.LinearScaleBase now must be imported and cannot be accessed off the Chart object
* Chart.offsetX
* Chart.offsetY
* Chart.outerRadius now lives on doughnut, pie, and polarArea controllers
* Chart.plugins was replaced with Chart.registry. Plugin defaults are now in Chart.defaults.plugins[id].
* Chart.plugins.register was replaced by Chart.register.
* Chart.PolarArea. New charts are created via new Chart and providing the appropriate type parameter
* Chart.prototype.generateLegend
* Chart.platform. It only contained disableCSSInjection. CSS is never injected in v3.
* Chart.PluginBase
* Chart.Radar. New charts are created via new Chart and providing the appropriate type parameter
* Chart.radiusLength
* Chart.scaleService was replaced with Chart.registry. Scale defaults are now in Chart.defaults.scales[type].
* Chart.Scatter. New charts are created via new Chart and providing the appropriate type parameter
* Chart.types
* Chart.Title was moved to Chart.plugins.title.\_element and made private
* Chart.Tooltip is now provided by the tooltip plugin. The positioners can be accessed from tooltipPlugin.positioners
* ILayoutItem.minSize

#### [**#**](#2jxsxqh) Removed from Dataset Controllers

* BarController.getDatasetMeta().bar
* DatasetController.addElementAndReset
* DatasetController.createMetaData
* DatasetController.createMetaDataset
* DoughnutController.getRingIndex

#### [**#**](#z337ya) Removed from Elements

* Element.getArea
* Element.height
* Element.hidden was replaced by chart level status, usable with getDataVisibility(index) / toggleDataVisibility(index)
* Element.initialize
* Element.inLabelRange
* Line.calculatePointY

#### [**#**](#3j2qqm3) Removed from Helpers

* helpers.addEvent
* helpers.aliasPixel
* helpers.arrayEquals
* helpers.configMerge
* helpers.findIndex
* helpers.findNextWhere
* helpers.findPreviousWhere
* helpers.extend. Use Object.assign instead
* helpers.getValueAtIndexOrDefault. Use helpers.resolve instead.
* helpers.indexOf
* helpers.lineTo
* helpers.longestText was made private
* helpers.max
* helpers.measureText was made private
* helpers.min
* helpers.nextItem
* helpers.niceNum
* helpers.numberOfLabelLines
* helpers.previousItem
* helpers.removeEvent
* helpers.roundedRect
* helpers.scaleMerge
* helpers.where

#### [**#**](#1y810tw) Removed from Layout

* Layout.defaults

#### [**#**](#4i7ojhp) Removed from Scales

* LinearScaleBase.handleDirectionalChanges
* LogarithmicScale.minNotZero
* Scale.getRightValue
* Scale.longestLabelWidth
* Scale.longestTextCache is now private
* Scale.margins is now private
* Scale.mergeTicksOptions
* Scale.ticksAsNumbers
* Scale.tickValues is now private
* TimeScale.getLabelCapacity is now private
* TimeScale.tickFormatFunction is now private

#### [**#**](#2xcytpi) Removed from Plugins (Legend, Title, and Tooltip)

* IPlugin.afterScaleUpdate. Use afterLayout instead
* Legend.margins is now private
* Legend onClick, onHover, and onLeave options now receive the legend as the 3rd argument in addition to implicitly via this
* Legend onClick, onHover, and onLeave options now receive a wrapped event as the first parameter. The previous first parameter value is accessible via event.native.
* Title.margins is now private
* The tooltip item's x and y attributes were replaced by element. You can use element.x and element.y or element.tooltipPosition() instead.

#### [**#**](#1ci93xb) Removal of Public APIs

The following public APIs were removed.

* getElementAtEvent is replaced with chart.getElementsAtEventForMode(e, 'nearest', { intersect: true }, false)
* getElementsAtEvent is replaced with chart.getElementsAtEventForMode(e, 'index', { intersect: true }, false)
* getElementsAtXAxis is replaced with chart.getElementsAtEventForMode(e, 'index', { intersect: false }, false)
* getDatasetAtEvent is replaced with chart.getElementsAtEventForMode(e, 'dataset', { intersect: true }, false)

#### [**#**](#3whwml4) Removal of private APIs

The following private APIs were removed.

* Chart.\_bufferedRender
* Chart.\_updating
* Chart.data.datasets[datasetIndex].\_meta
* DatasetController.\_getIndexScaleId
* DatasetController.\_getIndexScale
* DatasetController.\_getValueScaleId
* DatasetController.\_getValueScale
* Element.\_ctx
* Element.\_model
* Element.\_view
* LogarithmicScale.\_valueOffset
* TimeScale.getPixelForOffset
* TimeScale.getLabelWidth
* Tooltip.\_lastActive

### [**#**](#2bn6wsx) Renamed

The following properties were renamed during v3 development:

* Chart.Animation.animationObject was renamed to Chart.Animation
* Chart.Animation.chartInstance was renamed to Chart.Animation.chart
* Chart.canvasHelpers was merged with Chart.helpers
* Chart.elements.Arc was renamed to Chart.elements.ArcElement
* Chart.elements.Line was renamed to Chart.elements.LineElement
* Chart.elements.Point was renamed to Chart.elements.PointElement
* Chart.elements.Rectangle was renamed to Chart.elements.BarElement
* Chart.layoutService was renamed to Chart.layouts
* Chart.pluginService was renamed to Chart.plugins
* helpers.callCallback was renamed to helpers.callback
* helpers.drawRoundedRectangle was renamed to helpers.roundedRect
* helpers.getValueOrDefault was renamed to helpers.valueOrDefault
* LayoutItem.fullWidth was renamed to LayoutItem.fullSize
* Point.controlPointPreviousX was renamed to Point.cp1x
* Point.controlPointPreviousY was renamed to Point.cp1y
* Point.controlPointNextX was renamed to Point.cp2x
* Point.controlPointNextY was renamed to Point.cp2y
* Scale.calculateTickRotation was renamed to Scale.calculateLabelRotation
* Tooltip.options.legendColorBackgroupd was renamed to Tooltip.options.multiKeyBackground

#### [**#**](#qsh70q) Renamed private APIs

The private APIs listed below were renamed:

* BarController.calculateBarIndexPixels was renamed to BarController.\_calculateBarIndexPixels
* BarController.calculateBarValuePixels was renamed to BarController.\_calculateBarValuePixels
* BarController.getStackCount was renamed to BarController.\_getStackCount
* BarController.getStackIndex was renamed to BarController.\_getStackIndex
* BarController.getRuler was renamed to BarController.\_getRuler
* Chart.destroyDatasetMeta was renamed to Chart.\_destroyDatasetMeta
* Chart.drawDataset was renamed to Chart.\_drawDataset
* Chart.drawDatasets was renamed to Chart.\_drawDatasets
* Chart.eventHandler was renamed to Chart.\_eventHandler
* Chart.handleEvent was renamed to Chart.\_handleEvent
* Chart.initialize was renamed to Chart.\_initialize
* Chart.resetElements was renamed to Chart.\_resetElements
* Chart.unbindEvents was renamed to Chart.\_unbindEvents
* Chart.updateDataset was renamed to Chart.\_updateDataset
* Chart.updateDatasets was renamed to Chart.\_updateDatasets
* Chart.updateLayout was renamed to Chart.\_updateLayout
* DatasetController.destroy was renamed to DatasetController.\_destroy
* DatasetController.insertElements was renamed to DatasetController.\_insertElements
* DatasetController.onDataPop was renamed to DatasetController.\_onDataPop
* DatasetController.onDataPush was renamed to DatasetController.\_onDataPush
* DatasetController.onDataShift was renamed to DatasetController.\_onDataShift
* DatasetController.onDataSplice was renamed to DatasetController.\_onDataSplice
* DatasetController.onDataUnshift was renamed to DatasetController.\_onDataUnshift
* DatasetController.removeElements was renamed to DatasetController.\_removeElements
* DatasetController.resyncElements was renamed to DatasetController.\_resyncElements
* LayoutItem.isFullWidth was renamed to LayoutItem.isFullSize
* RadialLinearScale.setReductions was renamed to RadialLinearScale.\_setReductions
* RadialLinearScale.pointLabels was renamed to RadialLinearScale.\_pointLabels
* Scale.handleMargins was renamed to Scale.\_handleMargins

### [**#**](#3as4poj) Changed

The APIs listed in this section have changed in signature or behaviour from version 2.

#### [**#**](#1pxezwc) Changed in Scales

* Scale.getLabelForIndex was replaced by scale.getLabelForValue
* Scale.getPixelForValue now only requires one parameter. For the TimeScale that parameter must be millis since the epoch. As a performance optimization, it may take an optional second parameter, giving the index of the data point.

##### [**#**](#49x2ik5) Changed in Ticks

* Scale.afterBuildTicks now has no parameters like the other callbacks
* Scale.buildTicks is now expected to return tick objects
* Scale.convertTicksToLabels was renamed to generateTickLabels. It is now expected to set the label property on the ticks given as input
* Scale.ticks now contains objects instead of strings
* When the autoSkip option is enabled, Scale.ticks now contains only the non-skipped ticks instead of all ticks.
* Ticks are now always generated in monotonically increasing order

##### [**#**](#2p2csry) Changed in Time Scale

* getValueForPixel now returns milliseconds since the epoch

#### [**#**](#147n2zr) Changed in Controllers

##### [**#**](#3o7alnk) Core Controller

* The first parameter to updateHoverStyle is now an array of objects containing the element, datasetIndex, and index
* The signature or resize changed, the first silent parameter was removed.

##### [**#**](#23ckvvd) Dataset Controllers

* updateElement was replaced with updateElements now taking the elements to update, the start index, count, and mode
* setHoverStyle and removeHoverStyle now additionally take the datasetIndex and index

#### [**#**](#ihv636) Changed in Interactions

* Interaction mode methods now return an array of objects containing the element, datasetIndex, and index

#### [**#**](#32hioqz) Changed in Layout

* ILayoutItem.update no longer has a return value

#### [**#**](#1hmsyys) Changed in Helpers

All helpers are now exposed in a flat hierarchy, e.g., Chart.helpers.canvas.clipArea -> Chart.helpers.clipArea

##### [**#**](#41mghml) Canvas Helper

* The second parameter to drawPoint is now the full options object, so style, rotation, and radius are no longer passed explicitly
* helpers.getMaximumHeight was replaced by helpers.dom.getMaximumSize
* helpers.getMaximumWidth was replaced by helpers.dom.getMaximumSize
* helpers.clear was renamed to helpers.clearCanvas and now takes canvas and optionally ctx as parameter(s).
* helpers.retinaScale accepts optional third parameter forceStyle, which forces overriding current canvas style. forceRatio no longer falls back to window.devicePixelRatio, instead it defaults to 1.

#### [**#**](#2grqrue) Changed in Platform

* Chart.platform is no longer the platform object used by charts. Every chart instance now has a separate platform instance.
* Chart.platforms is an object that contains two usable platform classes, BasicPlatform and DomPlatform. It also contains BasePlatform, a class that all platforms must extend from.
* If the canvas passed in is an instance of OffscreenCanvas, the BasicPlatform is automatically used.
* isAttached method was added to platform.

#### [**#**](#vx1227) Changed in IPlugin interface

* All plugin hooks have unified signature with 3 arguments: chart, args and options. This means change in signature for these hooks: beforeInit, afterInit, reset, beforeLayout, afterLayout, beforeRender, afterRender, beforeDraw, afterDraw, beforeDatasetsDraw, afterDatasetsDraw, beforeEvent, afterEvent, resize, destroy.
* afterDatasetsUpdate, afterUpdate, beforeDatasetsUpdate, and beforeUpdate now receive args object as second argument. options argument is always the last and thus was moved from 2nd to 3rd place.
* afterEvent and beforeEvent now receive a wrapped event as the event property of the second argument. The native event is available via args.event.native.
* Initial resize is no longer silent. Meaning that resize event can fire between beforeInit and afterInit
* New hooks: install, start, stop, and uninstall
* afterEvent should notify about changes that need a render by setting args.changed to true. Because the args are shared with all plugins, it should only be set to true and not false.

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

←  [Usage](http://docs.google.com/docs/3.9.1/getting-started/usage.html)   [Accessibility](http://docs.google.com/docs/3.9.1/general/accessibility.html)  →