Settings

Edit New Page

Jump to bottom

Ray Luxembourg edited this page on Aug 1, 2019 · 26 revisions

Settings in Sigma

The configuration system in Sigma closely resembles the prototyping system found in JavaScript:

1. Each module in Sigma - renderers, controllers, graphs, anything that inherits from `sigma.classes.configurable` - can have a settings object with one or more keys (often instantiated within its constructors). When a function calls for a particular setting, it will request the setting from this key first.
2. If the module does not have its own settings object or the key is not found on it, it will then turn to the settings object on the `sigma` instance that is hosting it.
3. If the settings object on the `sigma` instance does not have it, it will finally turn to the global settings stored in the `sigma.settings` object, which are instantiated to their default values in `sigma.settings.js`. You can modify this object directly to affect all instances of `sigma` in your application.

Available settings list

Here is an exhaustive list of every settings that are recognized by default by the different sigma's core components:

Graph settings

Property	Type	Default Value	Description
clone	Boolean	`true`	If `true`, the objects used to instantiate nodes and edges will be shallow-copied to the graph. If `false`, the same objects will be used (allowing you to maintain reference to them).
immutable	Boolean	`true`	If `true`, the critical properties of nodes and edges (`id`, `source`, and `target`, as applicable) will be set as immutable.
verbose	Boolean	`true`	If `true`, Sigma will log its errors and warnings.

Renderers settings

Note: To change the background of sigma, you have to change the CSS rules of the container. Since the different layers are transparent (except of the drawn elements), the background of the container is the actual background of your display.

For everything else, here are the available settings:

Property	Type	Default Value	Description
defaultNodeType	String	`"def"`	Indicates the default value of the `type` property on nodes that do not specify it, and thus the sub-renderer that will be invoked to draw them.
defaultEdgeType	String	`"def"`	The default value on the `type` property of edges that do not specify it
defaultLabelColor	String	`"#000"`	The default color of node labels
defaultEdgeColor	String	`#000`	The default color of edges when **edgeColor** is set to `"default"`
defaultNodeColor	String	`#000`	The default color of nodes
defaultLabelSize	String	`14`	The default size of text used to draw node labels
edgeColor	String	`"source"`	When no `color` property is defined on the edge, determines how to determine the color. `"source"` and `"target"` use that node's color, respectively, while `"default"` just uses the **defaultEdgeColor** setting.
minArrowSize	Number	`0`	Defines the minimal edge's arrow display size.
font	String	`"arial"`	Defines the default font
fontStyle	String	`""`	A CSS descriptor for drawing the chosen font
labelColor	String	`"default"`	If set to `"node"`, the label will be drawn as the same computed color for the node. Otherwise, the default label color is used.
labelSize	String	`"fixed"`	Indicates how to choose the label's size. Avaliable values: `"fixed"`, `"proportional"`
labelSizeRatio	Number	`1`	The ratio between the font size of the label and the node size.

Property	Type	Default Value	Description
labelThreshold	Number	`8`	The minimum size a node must have on screen to see its label displayed. This does not affect hovering behavior.
webglOversamplingRatio	Number	`2`	The oversampling factor used in the WebGL renderer.

The following settings are dedicated to customize the appearance of hovered nodes:

Property	Type	Default Value	Description
borderSize	Number	`0`	The size of the border of hovered nodes.
defaultNodeBorderColor	String	`"#000"`	The default hovered node border's color.
hoverFont	String	`""`	The hovered node's label font. If an empty string, will inherit the `"font"` setting.
hoverFontStyle	String	`""`	The CSS style to apply to the hover font.
labelHoverShadow	String	`"default"`	Indicates how to choose the hovered node's shadow color. If `"node"`, will choose the color based on the node's shadow color.
labelHoverShadowColor	String	`"#000"`	The default color for the hovered node's shadow.
nodeHoverColor	String	`"node"`	Indicates how to choose the hovered node's color. If `"node"`, the node's normal color will be used. If `"default"`, the setting will be used.
defaultNodeHoverColor	String	`"#000"`	The default color of nodes when hovered.
labelHoverBGColor	String	`"default"`	Indicates how to choose the hovered node's background color. If `"node"`, the node's normal color will be used.
defaultHoverLabelBGColor	String	`"#fff"`	The default background color of hovered nodes' labels.
labelHoverColor	String	`"default"`	Indicates how to choose the hovered label's color. If `"node"`, the node's color will be used.
defaultLabelHoverColor	String	`"#000"`	The default text color of hovered labels.

Property	Type	Default Value	Description
singleHover	Boolean	`false`	If `true`, then only one node can be hovered at a time - as is possible on touch or on custom captors.

The following settings are dedicated to customize the appearance of hovered edges:

Property	Type	Default Value	Description
edgeHoverColor	String	`"edge"`	Indicates how to choose the edges' hover color. If `"default"`, will use the default color in the settings.
defaultEdgeHoverColor	String	`"#000"`	The default edge hover color.
edgeHoverSizeRatio	Number	`1`	Indicates how much to multiply edge size by when hovered
edgeHoverExtremities	Boolean	`false`	Indicates whether or not the edge's extremities (the nodes the edge connects) should be hovered with the edge.

If you wish, you can disable edge or node drawing entirely:

Property	Type	Default Value	Description
drawLabels	Boolean	`true`	Determines whether or not to draw node labels.
drawEdgeLabels	Boolean	`true`	Determines whether or not to draw edge labels.
drawEdges	Boolean	`true`	Determines whether or not to draw edges.
drawNodes	Boolean	`true`	Determines whether or not to draw nodes.

To speed up manipulation, Sigma is capable of batching edge rendering, distributing it to several frames. It is also capable of hiding the edges of the graph during drawing and animations. The following settings control these behaviors:

Property	Type	Default Value	Description
batchEdgesDrawing	Boolean	`false`	Indicates whether or not to spread edge rendering over multiple frames.
canvasEdgesBatchSize	Number	`500`	The number of edges to batch on the canvas renderer with batch drawing enabled.

Property	Type	Default Value	Description
webglEdgesBatchSize	Number	`1000`	The number of edges to batch on the WebGL renderer with batch drawing enabled.
hideEdgesOnMove	Boolean	`false`	If true, then edges won't draw during dragging or animations.

Rescale settings

Property	Type	Default Value	Description
scalingMode	String	`"inside"`	Indicates of to scale the graph relatively to its container. Available values: `"inside"`, `"outside"`
sideMargin	Number	`0`	The margin (in pixels) to keep around the graph

The following settings determine the size of the smallest and the biggest node / edges on the screen when the camera ratio is 0. This mapping makes easier to display the graph, avoiding too big nodes that take half of the screen, or too small ones that are not readable. If the min and max parameters are equals, then the minimal display size will be 0 and the maximal will be equal to the given value. And if they are both equal to 0, then there is no mapping, and the radius of the nodes will be their size.

Property	Type	Default Value
minEdgeSize	Number	`0.5`
maxEdgeSize	Number	`1`
minNodeSize	Number	`1`
maxNodeSize	Number	`8`

Captors settings

These settings configure Sigma's input captors

Property	Type	Default Value	Description
touchEnabled	Boolean	`true`	Indicates whether or not touch support is enabled
mouseEnabled	Boolean	`true`	Indicates whether or not mouse support is enabled
mouseWheelEnabled	Boolean	`true`	Indicates whether or not the mouse wheel is enabled

Property	Type	Default Value	Description
doubleClickEnabled	Boolean	`true`	Indicates whether or not the graph can be zoomed on double-click
eventsEnabled	Boolean	`true`	Indicates whether the captors dispatch events
zoomingRatio	Number	`1.7`	Determines how much the graph zooms with each turn of the mouse wheel
doubleClickZoomingRatio	Number	`2.2`	Determines how much the graph zooms with each double-click
zoomMin	Number	`0.0625`	The minimum zoom level
zoomMax	Number	`2`	The maximum zoom level
mouseZoomDuration	Number	`200`	The duration of a mouse wheel zoom animation in milliseconds
doubleClickZoomDuration	Number	`200`	The duration of a double-click zoom animation in milliseconds
mouseInertiaDuration	Number	`200`	The duration of mouse-drag inertia in milliseconds
mouseInertiaRatio	Number	`3`	The inertial power of the mouse
touchInertiaDuration	Number	`200`	The duration of touch-drag inertia in milliseconds
touchInertiaRatio	Number	`3`	The inertial power of the touch captor
doubleClickTimeout	Number	`300`	The time required for two clicks to be recognized as a double click
doubleTapTimeout	Number	`300`	The time required for two taps to be recognized as a double tap
dragTimeout	Number	`200`	The maximum time of dragging to trigger inertia

Global settings

Property	Type	Default Value	Description
autoResize	Boolean	`true`	If true, the instance will refresh itself whenever a `resize` event is dispatched from the window object.

Property	Type	Default Value	Description
autoRescale	Boolean OR ['nodePosition', 'nodeSize', 'edgeSize']	`true`	If true, the cameras will rescale automatically when Sigma refreshes (it will call the "rescale" middleware).
enableCamera	Boolean	`true`	If false, the camera will not move when `goTo` is called.
enableHovering	Boolean	`true`	If true, nodes can be hovered over to display special rendering and detail text.
enableEdgeHovering	Boolean	`false`	If set to `true`, the edges can be hovered to display special rendering and detail text. Currently, edge hovering only works on the Canvas renderer.
edgeHoverPrecision	Number	`5`	The size of the area around the edges to activate hovering.
rescaleIgnoreSize	Boolean	`false`	If set to `true`, the camera rescale routines will ignore node sizes to determine graph bounds.
skipErrors	Boolean	`false`	If `true`, Sigma will catch and ignore rendering errors.

Camera settings

The power degrees applied to the nodes / edges size relatively to the zooming level. Basically:

- `onScreenR = Math.pow(zoom, nodesPowRatio) * R`
- `onScreenT = T / Math.pow(zoom, edgesPowRatio)`

Property	Type	Default Value
nodesPowRatio	Number	`0.5`
edgesPowRatio	Number	`0.5`

Animations settings

Property	Type	Default Value	Description

Property	Type	Default Value	Description
animationsTime	Number	`200`	The default duration for animations created with the Animations plugin.