Package 'bpmnVisualizationR'

July 22, 2025

Type Package

Title Visualize Process Execution Data on 'BPMN' Diagrams

Version 0.5.0

Description To visualize the execution data of the processes on 'BPMN' (Business Process Model and Notation) diagrams, using overlays, style customization and interactions, with the 'bpmn-visualization' 'TypeScript' library.

License Apache License (== 2)

Copyright Bonitasoft S.A.

URL https://process-analytics.github.io/bpmn-visualization-R/, https://github.com/process-analytics/bpmn-visualization-R/

BugReports https://github.com/process-analytics/bpmn-visualization-R/issues/

Encoding UTF-8

RoxygenNote 7.2.3

Imports htmlwidgets, rlang, xml2

Suggests shiny, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Celine Souchet [aut, cre], Thomas Bouffard [aut]

Maintainer Celine Souchet <process.analytics.dev+CRAN@gmail.com>

Repository CRAN

Date/Publication 2023-09-18 14:50:02 UTC

Contents

`bpmnVisualizationR-shiny-output` *Shiny output binding for the* `bpmnVisualizationR` *'HTML' widget*

Description

Helper to create output function for using the `bpmnVisualizationR` 'HTML' widget within 'Shiny' applications and interactive 'Rmd' documents.

Usage

```
bpmnVisualizationROutput(outputId, width = "100%", height = "400px")
```

Arguments

`outputId`	output variable to read from
`width, height`	Must be a valid CSS unit (like `100%`, `400px`, `auto`) or a number, which will be coerced to a string and have `px` appended.

Value

An output function that enables the use of the `bpmnVisualizationR` widget within 'Shiny' applications.

`create_edge_style` *Create the style for BPMN edge*

Description

Use this function to create the correct style structure for the edge.

Usage

```
create_edge_style(
  elementIds,
  stroke_color = NULL,
  stroke_width = NULL,
  stroke_opacity = NULL,
  font_color = NULL,
  font_family = NULL,
  font_size = NULL,
  font_bold = NULL,
  font_italic = NULL,
  font_strike_through = NULL,
  font_underline = NULL,
  font_opacity = NULL,
  opacity = NULL
)
```

Arguments

`elementIds`	The IDs of the BPMN elements to style.
`stroke_color`	The stroke color for the element. It can be any HTML color name or HEX code, or special keywords.
`stroke_width`	The stroke width for the element, in pixels (1 to 50). • If the set value is less than 1, the used value is 1. • If the set value is greater than 50, the used value is 50. • To hide the stroke, set the `stroke_color` property to `none`.
`stroke_opacity`	The stroke opacity for the element, ranging from 0 to 100.
`font_color`	The font color for the element. It can be any HTML color name or HEX code, or special keywords.
`font_family`	The font family for the element.
`font_size`	The font size for the element, in pixels.
`font_bold`	Should the font be bold? (default: `FALSE`)
`font_italic`	Should the font be italic? (default: `FALSE`)
`font_strike_through`	Should the font have a strike-through style? (default: `FALSE`)
`font_underline`	Should the font be underlined? (default: `FALSE`)
`font_opacity`	The font opacity for the element, ranging from 0 to 100.
`opacity`	The opacity for the element, ranging from 0 to 100.

Value

A list representing the style for the BPMN edge.

Special keywords

`default` • This keyword allows you to reset a style property of the BPMN element to its initial value.
• When applied to color properties, it bypasses the color specified in the 'BPMN' source if 'BPMN in Color' support is enabled. Instead, it uses the color defined in the default style of the 'BPMN' element..

`inherit` Applies the value from the immediate parent element.

`none` No color (used to hide strokes). Not available for `font_color`.

`swimlane` Applies the value from the nearest parent element with type `ShapeBpmnElementKind.LANE` or `ShapeBpmnElementKind.POOL`.

Note

Opacity properties:

- If the set value is less than 0, the used value is 0.
- If the set value is greater than 100, the used value is 100.

Warning: `stroke_width` **property:**
Changing the stroke width of Activities may be misleading, as the default stroke widths have a meaning according to the 'BPMN' Specification.
For example, updating the stroke width of a task using the same value as the default stroke width of a Call Activity can be confusing.
In this case, you should also change another property, such as the stroke color, to allow the user to differentiate between them.

Examples

```
# Create a style with a blue stroke and a bold, red font.
edge_style <- create_edge_style(
  elementIds = list('id_1', 'id_2'),
  stroke_color = "blue",
  stroke_width = 2,
  font_color = "red",
  font_bold = TRUE
)
```

`create_gradient_fill` *Create a gradient fill style for an element*

Description

Create a gradient fill style for an element.

Usage

```
create_gradient_fill(direction, start_color, end_color)
```

Arguments

direction	The direction of the gradient (e.g., left-to-right, right-to-left, bottom-to-top, top-to-bottom). Taking the example of bottom-to-top, this means that the start color is at the bottom of the paint pattern and the end color is at the top, with a gradient between them.
start_color	The starting color of the gradient. It can be any HTML color name or HEX code, as well as special keywords such as inherit, none, swimlane.
end_color	The ending color of the gradient. It can be any HTML color name or HEX code, as well as special keywords such as inherit, none, swimlane.

Value

A gradient fill style object.

---

create_overlay *Create an overlay*

Description

An overlay can be added to existing elements in the diagram.

See the overlays argument in the display function.

Use this function to create the correct overlay structure.

Usage

```
create_overlay(elementId, label, style = NULL, position = NULL)
```

Arguments

elementId	The bpmn element id to which the overlay will be attached
label	'HTML' element to use as an overlay
style	The style of the overlay. Use create_overlay_style function to create the style object of an overlay and be aware of the enableDefaultOverlayStyle parameter in the display function.
position	The position of the overlay If the bpmn element where the overlay will be attached is a Shape, use overlay_shape_position. Otherwise, use overlay_edge_position.

Value

An overlay object

Examples

```
# Example 1: Create an overlay with shape position "top-left"
overlay_style <- create_overlay_style(
  font_color = 'DarkSlateGray',
  font_size = 23,
  fill_color = 'MistyRose',
  stroke_color = 'Red'
)

overlay <- create_overlay(
  "my-shape-id",
  "My Overlay Label",
  style = overlay_style,
  position = overlay_shape_position[1]
)

# Example 2: Create an overlay with edge position "end"
overlay_style <- create_overlay_style(
  font_color = 'DarkSlateGray',
  font_size = 23,
  fill_color = 'MistyRose',
  stroke_color = 'Red'
)

overlay <- create_overlay(
  "my-edge-id",
  "My Overlay Label",
  style = overlay_style,
  position = overlay_edge_position[2]
)
```

`create_overlay_style` *Create the style of an overlay*

Description

When adding an overlay to an existing element in a diagram, it's possible to customize its style.

Refer to the `style` parameter in the `create_overlay` function for more information.

Use this function to create the correct style structure for an overlay.

Usage

```
create_overlay_style(
  font_color = NULL,
  font_size = NULL,
  fill_color = NULL,
  stroke_color = NULL
)
```

Arguments

font_color	The font color of the overlay. It can be any HTML color name or HEX code.
font_size	The font size of the overlay. Specify a number in pixels.
fill_color	The color of the background of the overlay. It can be any HTML color name or HEX code.
stroke_color	The color of the stroke of the overlay. It can be any HTML color name or HEX code. If you don't want to display a stroke, you can set the color to: • transparent, • the same value as for the fill_color. This increases the padding/margin.

Value

The style object of the overlay

---

create_shape_style *Create the style for BPMN shape*

Description

Use this function to create the correct style structure for the shape.

Usage

```
create_shape_style(
  elementIds,
  stroke_color = NULL,
  stroke_width = NULL,
  stroke_opacity = NULL,
  font_color = NULL,
  font_family = NULL,
  font_size = NULL,
  font_bold = NULL,
  font_italic = NULL,
  font_strike_through = NULL,
  font_underline = NULL,
  font_opacity = NULL,
  opacity = NULL,
  fill_color = NULL,
  fill_opacity = NULL
)
```

Arguments

`elementIds`	The IDs of the BPMN elements to style.
`stroke_color`	The stroke color for the element. It can be any HTML color name or HEX code, or special keywords.
`stroke_width`	The stroke width for the element, in pixels (1 to 50). • If the set value is less than 1, the used value is 1. • If the set value is greater than 50, the used value is 50. • To hide the stroke, set the `stroke_color` property to `none`.
`stroke_opacity`	The stroke opacity for the element, ranging from 0 to 100.
`font_color`	The font color for the element. It can be any HTML color name or HEX code, or special keywords.
`font_family`	The font family for the element.
`font_size`	The font size for the element, in pixels.
`font_bold`	Should the font be bold? (default: `FALSE`)
`font_italic`	Should the font be italic? (default: `FALSE`)
`font_strike_through`	Should the font have a strike-through style? (default: `FALSE`)
`font_underline`	Should the font be underlined? (default: `FALSE`)
`font_opacity`	The font opacity for the element, ranging from 0 to 100.
`opacity`	The opacity for the element, ranging from 0 to 100.
`fill_color`	The fill color for the shape It can be any HTML color name or HEX code, special keywords, or a gradient create with `create_gradient_fill`.
`fill_opacity`	The fill opacity for the shape, ranging from 0 to 100.

Value

A list representing the style for the BPMN shape.

Special keywords

`default` • This keyword allows you to reset a style property of the BPMN element to its initial value.
• When applied to color properties, it bypasses the color specified in the 'BPMN' source if 'BPMN in Color' support is enabled. Instead, it uses the color defined in the default style of the 'BPMN' element..

`inherit` Applies the value from the immediate parent element.

`none` No color (used to hide strokes). Not available for `font_color`.

`swimlane` Applies the value from the nearest parent element with type `ShapeBpmnElementKind.LANE` or `ShapeBpmnElementKind.POOL`.

Note

Opacity properties:

- If the set value is less than 0, the used value is 0.
- If the set value is greater than 100, the used value is 100.

Warning: `stroke_width` **property:**
Changing the stroke width of Activities may be misleading, as the default stroke widths have a meaning according to the 'BPMN' Specification.
For example, updating the stroke width of a task using the same value as the default stroke width of a Call Activity can be confusing.
In this case, you should also change another property, such as the stroke color, to allow the user to differentiate between them.

See Also

`create_gradient_fill`

Examples

```
# Create a style with a blue stroke, red font, and green fill color.
shape_style <- create_shape_style(
  elementIds = list('id_1', 'id_2'),
  stroke_color = "blue",
  stroke_width = 2,
  font_color = "red",
  fill_color = "green"
)
```

---

display *Display 'BPMN' diagram in an 'HTML' Widget*

Description

Display 'BPMN' diagram based on 'BPMN' definition in 'XML' format

Usage

```
display(
  bpmnXML,
  overlays = NULL,
  enableDefaultOverlayStyle = TRUE,
  bpmnElementStyles = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

bpmnXML	A file name or 'XML' document or string in 'BPMN' 'XML' format
overlays	An element or a list of elements to be added to the diagram's existing elements. Use the create_overlay function to create an overlay object with content and a relative position.
enableDefaultOverlayStyle	If no style is set on an overlay, and this parameter is set to TRUE, the default style will be applied to the overlay. By default, enableDefaultOverlayStyle is set to TRUE.
bpmnElementStyles	a list of existing elements with their style to apply. Use the create_shape_style or create_edge_style functions to create the style of 'BPMN' elements.
width	A fixed width for the widget (in CSS units). The default value is NULL, which results in intelligent automatic sizing based on the widget's container.
height	A fixed height for the widget (in CSS units). The default value is NULL, which results in intelligent automatic sizing based on the widget's container.
elementId	The ID of the 'HTML' element to enclose the widget. Use an explicit element ID for the widget (rather than an automatically generated one). This is useful if you have other 'JavaScript' that needs to explicitly discover and interact with a specific widget instance.

Value

A bpmnVisualizationR widget that will intelligently print itself into 'HTML' in a variety of contexts including the 'R' console, within 'R Markdown' documents, and within 'Shiny' output bindings.

See Also

- create_overlay to create an overlay
- create_shape_style to create the structure style for the shape
- create_edge_style to create the structure style for the edge

Examples

```
# Load the BPMN file
bpmn_file <- system.file("examples/Order_Management.bpmn", package = "bpmnVisualizationR")

# Example 1: Display the BPMN diagram
bpmnVisualizationR::display(bpmn_file, width='auto', height='auto')

# Example 2: Display the BPMN diagram featuring overlays with their default positions and styles
overlays <- list(
  bpmnVisualizationR::create_overlay("start_event_1_1", "42"),
  bpmnVisualizationR::create_overlay("sequence_flow_1_1", "42"),
  bpmnVisualizationR::create_overlay("task_1_1", "9"),
  bpmnVisualizationR::create_overlay("sequence_flow_1_2", "8"),
  bpmnVisualizationR::create_overlay("call_activity_1_1", "7")
```

```
)

bpmnVisualizationR::display(
  bpmn_file,
  overlays,
  width='auto',
  height='auto'
)

# Example 3: Display the BPMN diagram featuring overlays using custom styles and positions
taskStyle <- bpmnVisualizationR::create_overlay_style(
  font_color = 'DarkSlateGray',
  font_size = 23,
  fill_color = 'MistyRose',
  stroke_color = 'Red'
)

flowStyle <- bpmnVisualizationR::create_overlay_style(
  font_color = 'WhiteSmoke',
  font_size = 19,
  fill_color = 'Teal',
  stroke_color = 'SpringGreen'
)

overlays <- list(
  bpmnVisualizationR::create_overlay("start_event_1_1", "42", position = "middle-left"),
  bpmnVisualizationR::create_overlay("sequence_flow_1_1", "42", flowStyle, "end"),
  bpmnVisualizationR::create_overlay("task_1_1", "9", taskStyle),
  bpmnVisualizationR::create_overlay("sequence_flow_1_2", "8"),
  bpmnVisualizationR::create_overlay("call_activity_1_1", "7")
)
bpmnVisualizationR::display(bpmn_file, overlays, width='auto', height='auto')

# Example 4: Display the BPMN diagram featuring overlays,
# but exclude their default styles and positions
overlays <- list(
  bpmnVisualizationR::create_overlay("start_event_1_1", "42", position = "middle-left"),
  bpmnVisualizationR::create_overlay("sequence_flow_1_1", "42", flowStyle, "end"),
  bpmnVisualizationR::create_overlay("task_1_1", "9", taskStyle, "bottom-right"),
  bpmnVisualizationR::create_overlay("sequence_flow_1_2", "8",  position = 'start')
)

bpmnVisualizationR::display(
  bpmn_file,
  overlays,
  enableDefaultOverlayStyle=FALSE,
  width='auto',
  height='auto'
)

# Example 5: Display the BPMN diagram featuring styling for BPMN elements
bpmnElementStyles <- list(
  bpmnVisualizationR::create_shape_style(
```

```
    elementIds = list("call_activity_1_1"),
    stroke_color = 'RoyalBlue',
    font_color = 'DarkOrange',
    font_family = 'Arial',
    font_size = 12,
    font_bold = TRUE,
    font_italic = TRUE,
    font_strike_through = TRUE,
    font_underline = TRUE,
    opacity = 75,
    fill_color = 'Yellow',
    fill_opacity = 50
  ),
  bpmnVisualizationR::create_edge_style(
    elementIds = list("sequence_flow_1_4"),
    stroke_color = 'DeepPink',
    stroke_width = 3,
    stroke_opacity = 70,
    font_color = 'ForestGreen',
    font_family = 'Courier New',
    font_size = 14,
    font_bold = TRUE,
    font_italic = TRUE,
    font_strike_through = FALSE,
    font_underline = FALSE,
    font_opacity = 80,
    opacity = 80
  )
)

bpmnVisualizationR::display(
  bpmn_file,
  bpmnElementStyles = bpmnElementStyles,
  width='auto',
  height='auto'
)
```

`overlay_edge_position` *The overlay positions on* `Edge`

Description

To specify the position when creating an overlay object that will be attached to BPMN `Edge` elements in the diagram.

Usage

```
overlay_edge_position
```

Format

An object of class `character` of length 3.

Details

Use these constants as the `position` argument in the `create_overlay` function.

Positions:

- `start`
- `end`
- `middle`

See Also

`create_overlay`

Examples

```
# Create an overlay at the starting point of an edge
overlay <- create_overlay(elementId = 1, label = "My label", position = overlay_edge_position[1])
```

`overlay_shape_position`

The overlay positions on `Shape`

Description

To specify the position when creating an overlay object that will be attached to BPMN `Shape` elements in the diagram.

Usage

```
overlay_shape_position
```

Format

An object of class `character` of length 8.

Details

Use these constants as the `position` argument in the `create_overlay` function.

Positions:

- `top-left`
- `top-right`

- top-center
- bottom-left
- bottom-right
- bottom-center
- middle-left
- middle-right

See Also

create_overlay

Examples

```
# Create an overlay at the top-left corner of a shape
overlay <- create_overlay(elementId = 1, label = "My label", position = overlay_shape_position[1])
```

---

renderBpmnVisualizationR

'Shiny' render binding for the bpmnVisualizationR *'HTML' widget*

Description

Helper to create render function for using the bpmnVisualizationR 'HTML' widget within 'Shiny' applications and interactive 'Rmd' documents.

Usage

```
renderBpmnVisualizationR(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	An expression that generates a bpmnVisualizationR 'HTML' widget
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Value

A render function that enables the use of the bpmnVisualizationR widget within 'Shiny' applications.

Index