

Package ‘blockr.dag’

July 30, 2026

Title A Directed Acyclic Graph Extension for 'blockr'

Version 0.1.5

Description Building on the docking layout manager provided by 'blockr.dock', this provides an extension that allows for visualizing and manipulating a 'blockr' board using a DAG-based user interface powered by the 'g6R' graph visualisation HTML widget.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports blockr.core (>= 0.1.3), blockr.dock (>= 0.1.2), shiny, g6R (>= 0.6.5), jsonlite, htmltools

Suggests knitr, rmarkdown, testthat (>= 3.0.0), roxy.shinylive, webshot2, quarto, cyclocomp, shinytest2, chromote, withr, colorspace

Config/testthat/edition 3

URL <https://bristolmyerssquibb.github.io/blockr.dag/>

VignetteBuilder quarto

NeedsCompilation no

Author David Granjon [aut, cre],
Nicolas Bennett [aut],
Christoph Sax [aut],
Bristol Myers Squibb [fnd]

Maintainer David Granjon <david@cynkra.com>

Repository CRAN

Date/Publication 2026-07-30 16:20:21 UTC

Contents

new_context_menu_entry	2
new_dag_extension	3
new_toolbar_item	4

Index**6**

new_context_menu_entry

*Context menu functions***Description**

Functions for creating and working with context menu entries.

Usage

```
new_context_menu_entry(
    name,
    js,
    action = NULL,
    condition = TRUE,
    id = tolower(gsub(" +", "_", name))
)
```

```
is_context_menu_entry(x)
```

```
context_menu_items(x)
```

Arguments

name	Name of the context menu entry.
js	JavaScript code to execute when the entry is selected.
action	Action to perform when the entry is selected.
condition	Condition to determine if the entry should be shown.
id	Unique identifier for the context menu entry. Inferred from name if not provided
x	Object

Details

`new_context_menu_entry()` Creates a new context menu entry with the specified name, JavaScript code, action function, and display condition.

`is_context_menu_entry()` Tests whether an object is a valid context menu entry.

`context_menu_items()` Generic function to extract context menu items from various objects like dock extensions, boards, or lists.

The `context_menu_items.dag_extension()` method provides the following actions:

- Create link - Creates connections between workflow nodes.
- Remove block - Removes individual blocks from the workflow.
- Remove link - Removes connections between workflow nodes.

- Append block - Adds a new block after the selected node.
- Create stack - Creates a new workflow stack.
- Remove stack - Removes an entire workflow stack.
- Edit stack - Opens stack editing interface.
- Add block - Adds a new block to the canvas.

Value

`new_context_menu_entry()` A context menu entry object of class "context_menu_entry" containing condition, action, and js functions, with name and id attributes.

`is_context_menu_entry()` TRUE if x is a context menu entry, FALSE otherwise.

`context_menu_items()` A list of context menu items for the given object.

<code>new_dag_extension</code>	<i>DAG extension</i>
--------------------------------	----------------------

Description

Visualizes the DAG (directed acyclic graph) underlying a board and provides UI elements to manipulate the board.

Usage

```
new_dag_extension(positions = NULL, ...)
```

Arguments

<code>positions</code>	Optional node positions overlaid on the board-derived nodes, as a named list keyed by block id, each element a list with numeric x and y (e.g. <code>list(a = list(x = 100, y = 200))</code>). Persisted across save / restore. Unknown or stale block ids are ignored. This handle is externally controllable: positions can be set programmatically through the board update lifecycle (<code>update(list(extensions = list(mod = list(<...>)))</code> which moves the corresponding nodes. Note: the auto-layout currently computes final node placement at cold start, so supplied positions are not yet honored over it (a follow-up will let positions pin over the layout).
<code>...</code>	Forwarded to <code>blockr.dock::new_dock_extension()</code> .

Value

A `dag_extension` object that extends the dock extension system for visualizing and manipulating DAG workflows.

Options

`blockr.dag.svg_renderer`: when `TRUE`, the DAG is rendered with the SVG renderer instead of the default canvas renderer. Canvas is the default because the SVG element reports `offsetWidth == 0`, which makes the underlying g-lite client/canvas coordinate scaling ignore the page zoom factor and desyncs hit-testing below 100% browser zoom. The SVG renderer keeps every element in the DOM, which the shinytest2 end-to-end tests need to query and screenshot, so they opt in via `AppDriver$new(options = list(blockr.dag.svg_renderer = TRUE))`.

<code>new_toolbar_item</code>	<i>Toolbar item functions</i>
-------------------------------	-------------------------------

Description

Functions for creating and working with toolbar items for the DAG interface.

Usage

```
new_toolbar_item(id, icon, js, action = NULL, tooltip = NULL)
```

```
is_toolbar_item(x)
```

```
toolbar_items(x)
```

Arguments

<code>id</code>	Unique identifier for the toolbar item.
<code>icon</code>	Name of an icon to show in the toolbar.
<code>js</code>	JavaScript code to execute when the entry is selected.
<code>action</code>	Action to perform when the entry is selected.
<code>tooltip</code>	Optional tooltip text for the entry.
<code>x</code>	Object

Details

`new_toolbar_item()` Creates a new toolbar item with the specified `id`, `icon`, JavaScript code, action function, and tooltip text.

`is_toolbar_item()` Tests whether an object is a valid toolbar item.

`toolbar_items()` Generic function to extract toolbar items from various objects like dock extensions, boards, or lists.

The `toolbar_items.dag_extension()` method provides the following actions:

- Zoom in - Increases the graph zoom level.
- Zoom out - Decreases the graph zoom level.
- Auto fit - Automatically fits the entire graph within the viewport.

- Layout - Reapplies the graph layout algorithm to reorganize nodes.
- Add block - Opens interface to add a new block to the workflow.
- Add stack - Creates a new workflow stack.
- Remove selected - Removes currently selected elements from the graph.

Value

`new_toolbar_item()` A toolbar item object of class "toolbar_item" containing action and js functions, with id, icon, and tooltip attributes.

`is_toolbar_item()` TRUE if x is a toolbar item, FALSE otherwise.

`toolbar_items()` A list of toolbar items for the given object.

Index

`blockr.dock::new_dock_extension()`, [3](#)

`context_menu_items`
 `(new_context_menu_entry)`, [2](#)

`is_context_menu_entry`
 `(new_context_menu_entry)`, [2](#)

`is_toolbar_item` `(new_toolbar_item)`, [4](#)

`new_context_menu_entry`, [2](#)

`new_dag_extension`, [3](#)

`new_toolbar_item`, [4](#)

`toolbar_items` `(new_toolbar_item)`, [4](#)