Package: bpmnR (via r-universe)

September 8, 2024

Type Package

Title Support for BPMN (Business Process Management Notation) Models

Version 0.1.1

Description Creating, rendering and writing BPMN diagrams <https://www.bpmn.org/>. Functionalities can be used to visualize and export BPMN diagrams created using the 'pm4py' and 'bupaRminer' packages. Part of the 'bupaR' ecosystem.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R(>= 3.0.0)

Imports DiagrammeR, DiagrammeRsvg, dplyr, htmltools, htmlwidgets, purrr, rvest, tidyr, uuid, DT, rlang, knitr, huxtable, stringr, readr, xml2, glue

VignetteBuilder knitr

Suggests rmarkdown

NeedsCompilation no

Author Alessio Nigro [aut], Gert Janssenswillen [cre], Ivan Esin [ctb], Hasselt University [cph]

Maintainer Gert Janssenswillen <gert.janssenswillen@uhasselt.be>

Date/Publication 2023-12-12 12:00:02 UTC

Repository https://gertjanssenswillen.r-universe.dev

RemoteUrl https://github.com/cran/bpmnR

RemoteRef HEAD

RemoteSha 6a5e817bc0233b57bd699a715609abb74d5c160f

Contents

---

bpmnR *bpmnR*

Description

Support for BPMN models

Author(s)

Maintainer: Gert Janssenswillen <gert.janssenswillen@uhasselt.be>

Authors:

- Alessio Nigro <alessionigro@hotmail.com>

Other contributors:

- Ivan Esin [contributor]
- Hasselt University [copyright holder]

---

calculate_CFC *Control Flow Complexity (CFC)* ———————————————-

---

Description

Control Flow Complexity (CFC) ———————————————-

Usage

```
calculate_CFC(bpmn)
```

Arguments

bpmn bpmn-object.

Value

Control-Flow-Complexity. Numeric vector of length 1

Examples

```
library(dplyr)
nodes <- tibble(id = "task", name = "Task name", objectType = "task",
gatewayDirection = NA)
events <- tibble(id = c("start","end"), name = c("Start event","End event"),
objectType = c("startEvent","endEvent"))
flows <- tibble(id = c("flow1","flow2"), name = c("flow1","flow2"),
sourceRef = c("start","task"), targetRef = c("task","end"),
objectType = c("sequenceFlow","sequenceFlow"))
model <- create_bpmn(nodes, flows, events)
calculate_CFC(model)
```

---

create_bpmn *Create BPMN object.*

Description

This creates a BPMN object by specifying a set of tasks, sequence flows, gateways, and a start and end event.

Usage

```
create_bpmn(nodes, flows, events)
```

Arguments

nodes	A data.frame of all nodes, with minimal columns id, name, objectType, gatewayDirection
flows	A data.frame of all flows, with minimal columns id, sourceRef, targetRef and objectType
events	A data.frame of all events, with minimal columns id, name, objectType

Value

A BPMN object as a list of data.frames for the BPMN elements and an XML document for the XML-based interchange format for the BPMN process.

Author(s)

Alessio Nigro

Examples

```
library(dplyr)
nodes <- tibble(id = "task", name = "Task name",
objectType = "task", gatewayDirection = NA)
events <- tibble(id = c("start","end"), name = c("Start event","End event"),
objectType = c("startEvent","endEvent"))
flows <- tibble(id = c("flow1","flow2"), name = c("flow1","flow2"),
sourceRef = c("start","task"), targetRef = c("task","end"),
objectType = c("sequenceFlow","sequenceFlow"))
create_bpmn(nodes, flows, events)
```

`create_xml` *Create XML document from BPMN object.*

Description

This creates an XML document based on a BPMN object.

Usage

```
create_xml(bpmn, ...)

## S3 method for class 'bpmn'
create_xml(bpmn, ...)
```

Arguments

`bpmn`	A BPMN object as a list of data.frames for the BPMN elements.
`...`	Additional arguments passed to methods.

Value

An XML document for the XML-based interchange format for the BPMN process.

Methods (by class)

- `create_xml(bpmn)`: Create xml

Author(s)

Alessio Nigro

Examples

```
library(dplyr)
nodes <- tibble(id = "task", name = "Task name", objectType = "task",
gatewayDirection = NA)
events <- tibble(id = c("start","end"), name = c("Start event","End event"),
objectType = c("startEvent","endEvent"))
flows <- tibble(id = c("flow1","flow2"), name = c("flow1","flow2"),
sourceRef = c("start","task"), targetRef = c("task","end"),
objectType = c("sequenceFlow","sequenceFlow"))
model <- create_bpmn(nodes, flows, events)
create_xml(model)
```

---

example_bpmn *Example bpmn input for create_bp*

Description

Contains 5 attributes (tasks, sequenceFlows, gateways, startEvent, endEvent) each one as a dataframe

Usage

```
bpmn_instance
```

Format

Eventlog containing 500 patient cases

---

print_bpmn *Print xml part of bpmn*

Description

Print xml part of bpmn

Usage

```
print_bpmn(x, ...)
```

Arguments

x	A bpmn object from create_bpmn function
...	Any additional arguments

Value

No return value, only print model.

Author(s)

Alessio Nigro

`render_bpmn` *Render BPMN diagram.*

Description

This renders a BPMN diagram based on a BPMN object.

Usage

```
render_bpmn(
  bpmn,
  viewer.suppress = FALSE,
  width = NULL,
  height = NULL,
  elementId = NULL,
  xml_version_number = "1.0",
  xml_encoding_declaration = "UTF-8",
  ...
)

## S3 method for class 'bpmn'
render_bpmn(
  bpmn,
  viewer.suppress = FALSE,
  width = NULL,
  height = NULL,
  elementId = NULL,
  xml_version_number = "1.0",
  xml_encoding_declaration = "UTF-8",
  ...
)
```

Arguments

`bpmn`	A BPMN object as a list of data.frames for the BPMN elements and an XML document for the XML-based interchange format for the BPMN process.
`viewer.suppress`	Never display the widget within the RStudio Viewer (useful for widgets that require a large amount of space for rendering). Defaults to FALSE.
`width`	Fixed width for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.
`height`	Fixed height for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.

elementId — Use an explicit element ID for the widget (rather than an automatically generated one). Useful if you have other JavaScript that needs to explicitly discover and interact with a specific widget instance.

xml_version_number — The version of the XML standard used.

xml_encoding_declaration — The character encoding used in the XML declaration. 'UTF-8' is the default encoding used.

... — Additional arguments passed to methods.

Value

Rendered BPMN model in htmlwidget.

Author(s)

Alessio Nigro

Examples

```
library(dplyr)
nodes <- tibble(id = "task", name = "Task name",
objectType = "task", gatewayDirection = NA)
events <- tibble(id = c("start","end"), name = c("Start event","End event"),
objectType = c("startEvent","endEvent"))
flows <- tibble(id = c("flow1","flow2"), name = c("flow1","flow2"),
sourceRef = c("start","task"), targetRef = c("task","end"),
objectType = c("sequenceFlow","sequenceFlow"))
model <- create_bpmn(nodes, flows, events)
render_bpmn(model)
```

---

render_bpmn-shiny *Shiny bindings for render_bpmn*

Description

Output and render functions for using render_bpmn within Shiny applications and interactive Rmd documents.

Usage

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

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

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.
expr	An expression that generates a render_bpmn
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

Rendered BPMN model in Shiny widget.

Rendered BPMN model in Shiny widget.

Author(s)

Alessio Nigro

---

write_bpmn *Write XML or HTML to disk.*

Description

This writes out both XML and normalised HTML. The default behavior will output the same format which was read. If you want to force output pass ‘option = "as_xml"‘ or ‘option = "as_html"‘ respectively.

Usage

```
write_bpmn(bpmn, file, ...)

## S3 method for class 'bpmn'
write_bpmn(bpmn, file, ...)
```

Arguments

bpmn	A BPMN object as a list of data.frames for the BPMN elements and an XML document for the XML-based interchange format for the BPMN process.
file	Path to file or connection to write to.
...	Additional arguments passed to methods.

Value

Writes file to system.

Methods (by class)

- `write_bpmn(bpmn)`: Write bpmn to .bpmn file

Author(s)

Alessio Nigro

Index