Package: jplantuml4r (via r-universe)

August 22, 2024

R6GeneratorNote Generated by r6-generator-maven-plugin: remove this line if you want to make manual changes and dont want them to get overwritten

Type Package

Title Jplantuml4r: R6 Java Wrapper Package

Version 0.0.0.9000

Description An R6 package wrapping java code in the org.github.io.github.terminological:jplantuml4r library. This library was generated by the r6-generator-maven-plugin.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Suggests here, knitr, rmarkdown, testthat

Imports rJava, R6, fs, rappdirs, utils, magrittr, rlang, xml2, stringr

URL `https://terminological.github.io/jplantuml4r/index.html`, `https://github.com/terminological/jplantuml4r`

BugReports `https://github.com/terminological/jplantuml4r/issues`

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

RemoteUrl https://github.com/terminological/jplantuml4r

RemoteRef 0.0.0.9000

RemoteSha ed229b27ef6452aca7c0fe04c6f3e708c5383539

Contents

`jplantuml4r-package` *jplantuml4r: Jplantuml4r: R6 Java Wrapper Package*

Description

An R6 package wrapping java code in the org.github.io.github.terminological:jplantuml4r library. This library was generated by the r6-generator-maven-plugin.

Version: 0.0.0.9000

Classes:

- `JavaApi`
- `PlantUML`

Author(s)

- Rob Challen rob.challen@bristol.ac.uk 0000-0002-5504-7768

`JavaApi` *Jplantuml4r: R6 Java Wrapper Package*

Description

An R6 package wrapping java code in the org.github.io.github.terminological:jplantuml4r library. This library was generated by the r6-generator-maven-plugin.

Version: 0.0.0.9000

Generated: 2024-04-24T15:45:25.865875868

Arguments

`logLevel` optional - the slf4j log level as a string - one of OFF (most specific, no logging), FATAL (most specific, little data), ERROR, WARN, INFO, DEBUG, TRACE (least specific, a lot of data), ALL (least specific, all data)

Usage

```
J = jplantuml4r::JavaApi$get(logLevel)
```

Package initialisation and control

- `JavaApi$installDependencies()`
- `JavaApi$rebuildDependencies()`
- `JavaApi$versionInformation()`
- `J = JavaApi$get(logLevel)`
- `J$changeLogLevel(logLevel)`
- `J$reconfigureLog(log4jproperties)`
- `J$printMessages()`

Package classes and static methods

- `J$PlantUML$new()`
- `J$PlantUML$savePlantUml(plantuml, outFile, format)`
- `J$PlantUML$supportedFormats()`

Package initialisation and control

Package method `JavaApi$installDependencies()`**:** This package level method checks for, and installs any dependencies needed for the running of the package. It is called automatically on first package load and so in general does not need to be used directly.

Usage:

```
jplantuml4r::JavaApi$installDependencies()
```

Arguments:

- none

Returns: nothing. called for side effects.

Package method `JavaApi$rebuildDependencies()`**:** This package level method removes existing dependencies and re-installs dependencies needed for the running of the package. It is called automatically on first package load and so in general does not need to be called.

Usage:

```
jplantuml4r::JavaApi$rebuildDependencies()
```

Arguments:

- none

Returns: nothing. called for side effects.

Package method `JavaApi$versionInformation()`**:** This package level method returns debugging version information for the package

Usage:

```
jplantuml4r::JavaApi$versionInformation()
```

Arguments:

- none

Returns: A list containing a set of versioning information about this package.

Package method `JavaApi$get()`**:** This is the main entry point for the package and the root of the Java API in this package. All classes defined in the package are made available as items under this root. The JavaApi object manages the communication between R and Java.

Usage:

```
J = jplantuml4r::JavaApi$get()
# package classes and functions are nested under the `J` api object.
```

Arguments:

- logLevel The desired verbosity of the package. One of "OFF", "FATAL", "ERROR", "WARN", "INFO", "DEBUG", "TRACE", "ALL".

Returns: A R6 jplantuml4r::JavaApi object containing the access point to the objects and functions defined in this package

Api method `J$changeLogLevel(logLevel)`**:** Once the package is initialised the log level can be changed to increase the level of messages from the api.

Usage:

```
J = jplantuml4r::JavaApi$get()
J$changeLogLevel("DEBUG")
```

Arguments:

- logLevel The desired verbosity of the package. One of "OFF", "FATAL", "ERROR", "WARN", "INFO", "DEBUG", "TRACE", "ALL".

Returns: nothing. used for side effects.

Api method `J$reconfigureLog(log4jproperties)`**:** Experimental / Advanced use: Once the package is initialised the log configureation can be changed to log to an external file for example.

Usage:

```
J = jplantuml4r::JavaApi$get()
prp = fs::path(getwd(),"log4j.properties")
if (fs::file_exists(prp)) {
 J$changeLogLevel(prp)
}
```

Arguments:

- log4jproperties a full path to a log4jproperies file

Returns: nothing. used for side effects.

Api method `J$printMessages()`**:** Experimental / Internal use: Messages from Java to R are queued and printed after each function call. It is unlikely that any will be not printed so in normal circumstances this function should do nothing.

Usage:

```
J = jplantuml4r::JavaApi$get()
J$printMessages()
```

Arguments:

- none

Returns: nothing. used for side effects.

Static methods and constructors

Method `PlantUML$new()`**:** the default no-args constructor

Usage:

```
J = jplantuml4r::JavaApi$get()
J$PlantUML$new()
```

Arguments:

- none

Returns: R6 PlantUML object:

Method `PlantUML$savePlantUml()`**:** Render a plantuml file to a png file

Usage:

```
J = jplantuml4r::JavaApi$get()
J$PlantUML$savePlantUml(plantuml, outFile, format)
# this method is also exposed as a package function:
jplantuml4r::save_plant_uml(plantuml, outFile, format)
```

Arguments:

- plantuml a plantuml file as a character vector. - (java expects a RCharacterVector)
- outFile the output filename - (defaulting to ‘tempfile()‘) - (java expects a RFile)
- format see supported_formats method - (defaulting to ‘’png’‘) - (java expects a RCharacter)

Returns: RFile: this java method returns the message that the object was created with

Method `PlantUML$supportedFormats()`**:** Get the list of supported formats for plantuml

Usage:

```
J = jplantuml4r::JavaApi$get()
J$PlantUML$supportedFormats()
# this method is also exposed as a package function:
jplantuml4r::supported_formats()
```

Arguments:

- none

Returns: RCharacterVector: a list of formats.

Author(s)

<rob.challen@bristol.ac.uk>

Examples

```
## -----------------------------------
## Check library dependencies for jplantuml4r
## -----------------------------------
jplantuml4r::JavaApi$installDependencies()

## -----------------------------------
## Construct a jplantuml4r Java API instance
## -----------------------------------

J = jplantuml4r::JavaApi$get()
# or a more verbose configuration
# J = jplantuml4r::JavaApi$get("DEBUG")


## -----------------------------------
## Method `J$PlantUML$new(...)`
## -----------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$PlantUML$new()
# or alternatively:
jplantuml4r::new()

## End(Not run)

## -----------------------------------
## Method `J$PlantUML$savePlantUml(...)`
## Aliased as `jplantuml4r::save_plant_uml(...)`
## -----------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$PlantUML$savePlantUml(plantuml, outFile, format)
# or alternatively:
jplantuml4r::save_plant_uml(plantuml, outFile, format)

## End(Not run)

## -----------------------------------
## Method `J$PlantUML$supportedFormats(...)`
## Aliased as `jplantuml4r::supported_formats(...)`
## -----------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$PlantUML$supportedFormats()
# or alternatively:
jplantuml4r::supported_formats()
```

```
## End(Not run)
```

PlantUML *PlantUML*

Description

PlantUML can generate a variety of diagrams from a DSL

Version: 0.0.0.9000

Generated: 2024-04-24T15:45:25.925080745

Details

PlantUML can generate a variety of diagrams from a DSL

Methods

Constructors:

- `J$PlantUML$new()`

Class methods:

- none
- `instance$clone()`
- `instance$print()`

Constructor `new()`**:** the default no-args constructor

Usage:

```
J = jplantuml4r::JavaApi$get()
instance = J$PlantUML$new();
```

Arguments:

- none

Returns: the new R6 PlantUML object

Examples

```
## ------------------------------------
## Construct new instance of PlantUML
## ------------------------------------
## Not run:
J = jplantuml4r::JavaApi$get()
# appropriate parameter values must be provided
instance = J$PlantUML$new()

## End(Not run)
```

`save_plant_uml` *Render a plantuml file to a png file*

Description

no description

Usage

```
save_plant_uml(
 plantuml,
 outFile,
 format
)
```

Arguments

`plantuml` plantuml a plantuml file as a character vector. - (java expects a RCharacterVector)

`outFile` outFile the output filename - (defaulting to ‘tempfile()‘) - (java expects a RFile)

`format` format see supported_formats method - (defaulting to ‘’png’‘) - (java expects a RCharacter)

Value

RFile: this java method returns the message that the object was created with

`supported_formats` *Get the list of supported formats for plantuml*

Description

no description

Usage

```
supported_formats()
```

Value

RCharacterVector: a list of formats.

Index