Package: roogledocs (via r-universe)

February 23, 2025

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 R Wrapper For Googledocs Java Library

Version 0.5.0

Description Programmatically substitute images, data and tables into a google doc or presentation. R library to perform limited interactions with google docs and slides in R via the Java API library. The purpose being to support google docs as a platform for interactive development and documentation of data analysis in R for scientific publication, although it is not limited to this purpose. The workflow supported is a parallel documentation and analysis where a team of people are working collaboratively on documentation, whilst at the same time analysis is being performed and results updated repeatedly as a result of new data. In this environment updating numeric results, tabular data and figures in word documents manually becomes annoying. With roogledocs you can automate this a bit like a RMarkdown document, but with the added benefit that the content can be updated independently of the analysis, by the wider team.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Suggests here, knitr, rmarkdown, testthat

Imports ggplot2, ragg, tidyr, dplyr, rJava, R6, fs, rappdirs, utils, magrittr, rlang, xml2, stringr

URL https://terminological.github.io/roogledocs/, https://github.com/terminological/roogledocs

BugReports https://github.com/terminological/roogledocs/issues

RoxygenNote 7.3.1

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev default-jdk libicu-dev libjpeg-dev libpng-dev libtiff-dev libxml2-dev

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

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

RemoteRef 0.5.0

RemoteSha b51fd77a7ddcebb8b496e039120c8dc38a7a8ec1

Contents

`roogledocs-package` *roogledocs: R Wrapper For Googledocs Java Library*

Description

Programmatically substitute images, data and tables into a google doc or presentation. R library to perform limited interactions with google docs and slides in R via the Java API library. The purpose being to support google docs as a platform for interactive development and documentation of data analysis in R for scientific publication, although it is not limited to this purpose. The workflow supported is a parallel documentation and analysis where a team of people are working collaboratively on documentation, whilst at the same time analysis is being performed and results updated repeatedly as a result of new data. In this environment updating numeric results, tabular data and figures in word documents manually becomes annoying. With roogledocs you can automate this a bit like

a RMarkdown document, but with the added benefit that the content can be updated independently of the analysis, by the wider team.

Version: 0.5.0

Classes:

- `JavaApi`
- `RoogleDocs`
- `RoogleSlides`

Author(s)

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

`as.long_format_table` *Convert a table to long format*

Description

Converts a square display table format to a long format suitable for applying as a sequence of formatting operations in a google doc or as a ggplot. Currently only plain dataframes and huxtables are supported but flextables look very doable. Only a limited subset of formatting features is implemented at present as supported by roogledocs. The output format is a simple dataframe with the following columns:

Usage

```
as.long_format_table(table, ...)
```

Arguments

`table`	the input table (e.g. a huxtable)
`...`	passed onto subclass methods

Details

- Character: label - non blank text (a single space is OK but not an empty string) - Integer: row - must be an integer, 1-based from top left - Integer: col - must be an integer, 1-based from top left - Integer: rowSpan - must be an integer, minimum value 1 - Integer: colSpan - must be an integer, minimum value 1 - Character: fontName - font name as seen in font drop down of google docs e.g "Roboto","Arial","Times New Roman", unrecognised values will be displayed as Arial - Character: fontFace - one of "bold", "bold.italic", "italic", "plain" - Numeric: fontSize - in points - Character: fillColour - as a hex string e.g. "#aaaaaa". N.b. British English spelling (sorry) - Numeric: leftBorderWeight - border weight in points - minimum size that appears in google docs is 0.5 - Numeric: rightBorderWeight - Numeric: topBorderWeight - Numeric: bottomBorderWeight - Character: alignment - one of "START","CENTER","END" - Character: valignment - one of "TOP","MIDDLE","BOTTOM"

It also has an attribute ‘colWidths‘ which is a vector the same length as the width of the table containing the relative widths of the columns. The overall table width is decided on rendering.

So not supported at the moment are border line types, border colours, control of padding, row height control, alignment on a decimal point, complex content / markup in cells.

Value

a format that is considered valid for roogledocs::RoogleDocs$updateTable()

`citation_styles` *List the supported citation styles*

Description

no description

Usage

```
citation_styles()
```

Value

RCharacterVector: a vector of csl style names

`delete_document` *Deletes a google document by name.*

Description

no description

Usage

```
delete_document(
 docName,
 areYouSure,
 tokenDirectory,
 disabled
)
```

Arguments

docName	docName - the name of a document to delete. must be an exact and unique match. - (java expects a RCharacter)
areYouSure	areYouSure - a boolean check. - (defaulting to 'utils::askYesNo(paste0('Are you sure ...') - (java expects a RLogical)
tokenDirectory	tokenDirectory - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

void: nothing, called for side efffects

---

delete_slides *Deletes a google slides by name.*

Description

no description

Usage

```
delete_slides(
 docName,
 areYouSure,
 tokenDirectory,
 disabled
)
```

Arguments

docName	docName - the name of a document to delete. must be an exact and unique match. - (java expects a RCharacter)
areYouSure	areYouSure - a boolean check. - (defaulting to 'utils::askYesNo(paste0('Are you sure ...') - (java expects a RLogical)
tokenDirectory	tokenDirectory - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

void: nothing, called for side efffects

doc_by_id *Get a document by id or sharing link.*

Description

no description

Usage

```
doc_by_id(
 shareUrlOrDocId,
 tokenDirectory,
 disabled
)
```

Arguments

shareUrlOrDocId	shareUrlOrDocId the url from clicking a share button in google docs or an id from searchForDocuments() method - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleDocs object: itself - a fluent method

doc_by_name *Get a document by name or create a blank document if missing.*

Description

no description

Usage

```
doc_by_name(
 title,
 tokenDirectory,
 disabled
)
```

Arguments

title	title a document title. If there is an exact match in google drive then that document will be used - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleDocs object: itself - a fluent method

---

doc_from_template *Get a document by name or create one from a template if missing.*

Description

no description

Usage

```
doc_from_template(
 title,
 templateUri,
 tokenDirectory,
 disabled
)
```

Arguments

title	title a document title. If there is an exact match in google drive then that document will be used otherwise a new one will be created. - (java expects a RCharacter)
templateUri	templateUri the share link (or document id) of a template google document - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleDocs object: itself - a fluent method

---

ggplot_to_png *Plot a ggplot object to a temporary file as a png*

Description

In ‘roogledocs‘ a path to a local png is needed as input to tagged image commands. This is a simple wrapper function to save a png using ‘ragg‘ with DPI fixed at 300 to a temporary directory and return the file path.

Usage

```
ggplot_to_png(plot, width, height)
```

Arguments

plot	a ggplot or patchwork object
width	the width of the output in inches
height	the height of the output in inches

Value

a full path to a png in a temp directory

Examples

```
if (FALSE) {
  doc = roogledocs::slides_by_name("roogledocs-demo")
  g = ggplot(diamonds, aes(x=carat,y=price, colour=color))+geom_point()
  figure_1 = ggplot_to_png(g, width=4, height=4)
  doc$updateTaggedImage(figure_1)

  # same as:
  # doc$updateTaggedImage(ggplot_to_png(plot,4,4), tagName = "figure_1")

}
```

JavaApi *R Wrapper For Googledocs Java Library*

Description

Programmatically substitute images, data and tables into a google doc or presentation. R library to perform limited interactions with google docs and slides in R via the Java API library. The purpose being to support google docs as a platform for interactive development and documentation of data analysis in R for scientific publication, although it is not limited to this purpose. The workflow supported is a parallel documentation and analysis where a team of people are working collaboratively on documentation, whilst at the same time analysis is being performed and results updated repeatedly as a result of new data. In this environment updating numeric results, tabular data and figures in word documents manually becomes annoying. With roogledocs you can automate this a bit like a RMarkdown document, but with the added benefit that the content can be updated independently of the analysis, by the wider team.

Version: 0.5.0

Generated: 2024-04-27T13:56:10.506316288

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 = roogledocs::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$RoogleDocs$new(tokenDirectory, disabled)`
- `J$RoogleDocs$reauth(tokenDirectory)`
- `J$RoogleDocs$docById(shareUrlOrDocId, tokenDirectory, disabled)`

- J$RoogleDocs$docByName(title, tokenDirectory, disabled)
- J$RoogleDocs$docFromTemplate(title, templateUri, tokenDirectory, disabled)
- J$RoogleDocs$searchForDocuments(titleMatch, tokenDirectory)
- J$RoogleDocs$deleteDocument(docName, areYouSure, tokenDirectory, disabled)
- J$RoogleDocs$citationStyles()
- J$RoogleSlides$new(tokenDirectory, disabled)
- J$RoogleSlides$slidesById(shareUrlOrDocId, tokenDirectory, disabled)
- J$RoogleSlides$slidesByName(title, tokenDirectory, disabled)
- J$RoogleSlides$slidesFromTemplate(title, templateUri, tokenDirectory, disabled)
- J$RoogleSlides$searchForSlides(titleMatch, tokenDirectory)
- J$RoogleSlides$deleteSlides(docName, areYouSure, tokenDirectory, disabled)

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:

```
roogledocs::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:

```
roogledocs::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:

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

Arguments:

- none

Returns: nothing. used for side effects.

Static methods and constructors

Method `RoogleDocs$new()`**:** Create a Roogledocs object for managing the interaction.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$new(tokenDirectory, disabled)
```

Arguments:

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleDocs object:

Method `RoogleDocs$reauth()`**:** Re-authenticate roogledocs library

Re-authenticate the service deleting the existing OAuth tokens may be helpful if there is some problem.
Generally this is only be needed if application permission updates are needed in which case the directory can be manually deleted anyway, or if you want to switch google user without using a different tokenDirectory.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$reauth(tokenDirectory)
# this method is also exposed as a package function:
roogledocs::reauth(tokenDirectory)
```

Arguments:

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

Returns: R6 RoogleDocs object: a new RoogleDocs instance without an active document

Method `RoogleDocs$docById()`**:** Get a document by id or sharing link.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$docById(shareUrlOrDocId, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::doc_by_id(shareUrlOrDocId, tokenDirectory, disabled)
```

Arguments:

shareUrlOrDocId the url from clicking a share button in google docs or an id from searchForDocuments() metho
- (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `RoogleDocs$docByName()`**:** Get a document by name or create a blank document if missing.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$docByName(title, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::doc_by_name(title, tokenDirectory, disabled)
```

Arguments:

title a document title. If there is an exact match in google drive then that document will be used
- (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `RoogleDocs$docFromTemplate()`**:** Get a document by name or create one from a template if missing.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$docFromTemplate(title, templateUri, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::doc_from_template(title, templateUri, tokenDirectory, disabled)
```

Arguments:

title a document title. If there is an exact match in google drive then that document will be used otherwise a new c
- (java expects a RCharacter)

templateUri the share link (or document id) of a template google document - (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `RoogleDocs$searchForDocuments()`**:** Search for documents with the given title

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$searchForDocuments(titleMatch, tokenDirectory)
# this method is also exposed as a package function:
roogledocs::search_for_documents(titleMatch, tokenDirectory)
```

Arguments:

titleMatch a string to be searched for as an approximate match. All results will be retrieved with document ids. - (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul - (java expects a RCharacter)

Returns: RDataframe: a dataframe containing id and name columns

Method `RoogleDocs$deleteDocument()`**:** Deletes a google document by name.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$deleteDocument(docName, areYouSure, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::delete_document(docName, areYouSure, tokenDirectory, disabled)
```

Arguments:

docName - the name of a document to delete. must be an exact and unique match. - (java expects a RCharacter)

areYouSure - a boolean check. - (defaulting to 'utils::askYesNo(paste0('Are you sure ...') - (java expects a RLogical)

tokenDirectory - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

disabled - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Returns: void: nothing, called for side efffects

Method `RoogleDocs$citationStyles()`**:** List the supported citation styles

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleDocs$citationStyles()
# this method is also exposed as a package function:
roogledocs::citation_styles()
```

Arguments:

- none

Returns: RCharacterVector: a vector of csl style names

Method `RoogleSlides$new()`**:** Create a RoogleSlides object for managing the interaction.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$new(tokenDirectory, disabled)
```

Arguments:

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleSlides object:

Method `RoogleSlides$slidesById()`**:** Get a document by id or sharing link.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$slidesById(shareUrlOrDocId, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::slides_by_id(shareUrlOrDocId, tokenDirectory, disabled)
```

Arguments:

shareUrlOrDocId the url from clicking a share button in google slides or an id from searchForDocuments() metho
- (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `RoogleSlides$slidesByName()`**:** Get a document by name or create a blank document
if missing.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$slidesByName(title, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::slides_by_name(title, tokenDirectory, disabled)
```

Arguments:

title a document title. If there is an exact match in google drive then that document will be used
- (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `RoogleSlides$slidesFromTemplate()`**:** Get a document by name or create one from a template if missing.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$slidesFromTemplate(title, templateUri, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::slides_from_template(title, templateUri, tokenDirectory, disabled)
```

Arguments:

title a document title. If there is an exact match in google drive then that document will be used otherwise a new c - (java expects a RCharacter)

templateUri the share link (or document id) of a template google document - (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul - (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b - (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `RoogleSlides$searchForSlides()`**:** Search for documents with the given title

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$searchForSlides(titleMatch, tokenDirectory)
# this method is also exposed as a package function:
roogledocs::search_for_slides(titleMatch, tokenDirectory)
```

Arguments:

titleMatch a string to be searched for as an approximate match. All results will be retrieved with document ids. - (java expects a RCharacter)

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul - (java expects a RCharacter)

Returns: RDataframe: a dataframe containing id and name columns

Method `RoogleSlides$deleteSlides()`**:** Deletes a google slides by name.

Usage:

```
J = roogledocs::JavaApi$get()
J$RoogleSlides$deleteSlides(docName, areYouSure, tokenDirectory, disabled)
# this method is also exposed as a package function:
roogledocs::delete_slides(docName, areYouSure, tokenDirectory, disabled)
```

Arguments:

docName - the name of a document to delete. must be an exact and unique match. - (java expects a RCharacter)

areYouSure - a boolean check. - (defaulting to 'utils::askYesNo(paste0('Are you sure ...') - (java expects a RLogical)

tokenDirectory - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

disabled - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Returns: void: nothing, called for side efffects

Author(s)

<rob.challen@bristol.ac.uk>

Examples

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

## ------------------------------------
## Construct a roogledocs Java API instance
## ------------------------------------

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


## ------------------------------------
## Method `J$RoogleDocs$new(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$new(tokenDirectory, disabled)
# or alternatively:
roogledocs::new(tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$reauth(...)`
## Aliased as `roogledocs::reauth(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$reauth(tokenDirectory)
# or alternatively:
roogledocs::reauth(tokenDirectory)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$docById(...)`
```

```
## Aliased as `roogledocs::doc_by_id(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$docById(shareUrlOrDocId, tokenDirectory, disabled)
# or alternatively:
roogledocs::doc_by_id(shareUrlOrDocId, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$docByName(...)`
## Aliased as `roogledocs::doc_by_name(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$docByName(title, tokenDirectory, disabled)
# or alternatively:
roogledocs::doc_by_name(title, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$docFromTemplate(...)`
## Aliased as `roogledocs::doc_from_template(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$docFromTemplate(title, templateUri, tokenDirectory, disabled)
# or alternatively:
roogledocs::doc_from_template(title, templateUri, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$searchForDocuments(...)`
## Aliased as `roogledocs::search_for_documents(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$searchForDocuments(titleMatch, tokenDirectory)
# or alternatively:
roogledocs::search_for_documents(titleMatch, tokenDirectory)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$deleteDocument(...)`
## Aliased as `roogledocs::delete_document(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$deleteDocument(docName, areYouSure, tokenDirectory, disabled)
```

```
# or alternatively:
roogledocs::delete_document(docName, areYouSure, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleDocs$citationStyles(...)`
## Aliased as `roogledocs::citation_styles(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleDocs$citationStyles()
# or alternatively:
roogledocs::citation_styles()

## End(Not run)

## ------------------------------------
## Method `J$RoogleSlides$new(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$new(tokenDirectory, disabled)
# or alternatively:
roogledocs::new(tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleSlides$slidesById(...)`
## Aliased as `roogledocs::slides_by_id(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$slidesById(shareUrlOrDocId, tokenDirectory, disabled)
# or alternatively:
roogledocs::slides_by_id(shareUrlOrDocId, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleSlides$slidesByName(...)`
## Aliased as `roogledocs::slides_by_name(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$slidesByName(title, tokenDirectory, disabled)
# or alternatively:
roogledocs::slides_by_name(title, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
```

```
## Method `J$RoogleSlides$slidesFromTemplate(...)`
## Aliased as `roogledocs::slides_from_template(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$slidesFromTemplate(title, templateUri, tokenDirectory, disabled)
# or alternatively:
roogledocs::slides_from_template(title, templateUri, tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `J$RoogleSlides$searchForSlides(...)`
## Aliased as `roogledocs::search_for_slides(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$searchForSlides(titleMatch, tokenDirectory)
# or alternatively:
roogledocs::search_for_slides(titleMatch, tokenDirectory)

## End(Not run)

## ------------------------------------
## Method `J$RoogleSlides$deleteSlides(...)`
## Aliased as `roogledocs::delete_slides(...)`
## ------------------------------------
## Not run:
# no example given - appropriate parameter values must be provided:
J$RoogleSlides$deleteSlides(docName, areYouSure, tokenDirectory, disabled)
# or alternatively:
roogledocs::delete_slides(docName, areYouSure, tokenDirectory, disabled)

## End(Not run)
```

`reauth` *Re-authenticate roogledocs library*

Description

Re-authenticate the service deleting the existing OAuth tokens may be helpful if there is some problem.

Generally this is only be needed if application permission updates are needed in which case the directory can be manually deleted anyway, or if you want to switch google user without using a different tokenDirectory.

Usage

```
reauth(
 tokenDirectory
)
```

Arguments

`tokenDirectory`	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

Value

R6 RoogleDocs object: a new RoogleDocs instance without an active document

---

`RoogleDocs` *RoogleDocs*

Description

Programmatically substitute images, data and tables into a google doc.

Version: 0.5.0

Generated: 2024-04-27T13:56:10.603218214

Arguments

`tokenDirectory`	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
`disabled`	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Details

R library to perform limited interactions with google docs (and maybe one day slides) in R via the Java API library. The purpose being to support google docs as a platform for interactive development and documentation of data analysis in R for scientific publication, although it is not limited to this purpose. The workflow supported is a parallel documentation and analysis where a team of people are working collaboratively on documentation, whilst at the same time analysis is being performed and results updated repeatedly as a result of new data. In this environment updating numeric results, tabular data and figures in word documents manually becomes annoying. With roogledocs you can automate this a bit like a RMarkdown document, but with the added benefit that the content can be updated independently of the analysis, by the wider team.

Methods

Constructors:

- `J$RoogleDocs$new(tokenDirectory, disabled)`

Class methods:

- instance$enable()
- instance$disable()
- instance$getName(suffix)
- instance$tagsDefined()
- instance$updateTaggedText(text, tagName)
- instance$updateTaggedImage(absoluteFilePath, tagName, dpi, keepUpload)
- instance$updateTaggedTable(longFormatTable, tagName, colWidths, tableWidthInches)
- instance$revertTags()
- instance$removeTags(confirm)
- instance$updateTable(longFormatTable, tableIndex, colWidths, tableWidthInches)
- instance$updateFigure(absoluteFilePath, figureIndex, dpi, keepUpload)
- instance$saveAsPdf(absoluteFilePath, uploadCopy)
- instance$makeCopy(newName)
- instance$delete(areYouSure)
- instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)
- instance$appendText(text, style)
- instance$appendFormattedParagraph(formattedTextDf)
- instance$updateCitations(bibTexPath, citationStyle)
- instance$clone()
- instance$print()

Constructor `new()`**:** Create a Roogledocs object for managing the interaction.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
```

Arguments:

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: the new R6 RoogleDocs object

Method `enable()`**:** Enables roogledocs method calls for this document.
It is likely one of 'withDocument()', 'findOrCreateDocument()' or 'findOrCloneTemplate()' methods will be needed to specify the document.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$enable()
```

Arguments:

- none

Returns: R6 RoogleDocs object: itself - a fluent method

Method `disable()`**:** Disables roogledocs temporarily for this document.
While disabled all calls to roogledocs will silently abort.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$disable()
```

Arguments:

- none

Returns: R6 RoogleDocs object: itself - a fluent method

Method `getName()`**:** Return the name of the document

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$getName(suffix)
```

Arguments:

suffix an additional suffix to add to the name - (defaulting to "") - (java expects a RCharacter)

Returns: String:

Method `tagsDefined()`**:** List all tags
Finds tags defined in the current document

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$tagsDefined()
```

Arguments:

- none

Returns: RDataframe: a dataframe containing tag and count columns

Method `updateTaggedText()`**:** Replace tags for text
Substitutes all occurrences of {{tag-name}} with the text parameter. If the tag name is not found in the document it is inserted at the end in a section labelled "Unmatched tags:". From there it can be cut and pasted into the right place.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateTaggedText(text, tagName)
```

Arguments:

text the value to replace the tag with (e.g. a result from analysis) (cannot be empty) - (java expects a RCharacter)

tagName the tag name - (defaulting to 'deparse(substitute(text))') - (java expects a RCharacter)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `updateTaggedImage()`**:** Replace a tag with an image.

Substitutes all occurrences of {{tag-name}} with an image from the local storage.

The image is uploaded to your google drive as a temporary file, and briefly made publicly readable. From there it is inserted into the google doc, and one completed the temporary file deleted from your google drive. Insertion is done using the dimensions of the existing image (if present) or the PNG dimensions if not. Creating the image with the correct dimensions (and providing the dpi if not 300) is important.

If the tag is missing from the document the image is inserted at the end (and tagged). From there it can be cut and pasted to the correct place.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateTaggedImage(absoluteFilePath, tagName, dpi, keepUpload)
```

Arguments:

absoluteFilePath a file path to an png image file. - (java expects a RCharacter)

tagName the tag name - (defaulting to 'deparse(substitute(absoluteFilePath))') - (java expects a RCharacter)

dpi the dots per inch of the image in the document (defaults to 300) - (defaulting to '300') - (java expects a RNumeric)

keepUpload keep the uploaded image as a supplementary file in the same directory as the google doc - (defaulting - (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `updateTaggedTable()`**:** Replace a tag with a table.

Substitutes a unique occurrences of {{tag-name}} with a table. The tag should either be in the text of the document or as the first entry in the first cell of a table. Once inserted the table is tagged using a zero width character as the very first item in the first cell. This will be removed if 'removeTags()' is called.

If the tag is missing the table is inserted at the end of the document where it can be cut and pasted to the correct place.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateTaggedTable(longFormatTable, tagName, colWidths, tableWidthInches)
```

Arguments:

longFormatTable A dataframe consisting of the table content and formatting indexed by row and column. at a mi
- (java expects a RDataframe)

tagName the tag name - (defaulting to 'deparse(substitute(longFormatTable))') - (java ex-
pects a RCharacter)

colWidths A vector including the relative length of each column. This can be left out if longFormatTable comes fr
- (java expects a RNumericVector)

tableWidthInches The final width of the table in inches (defaults to a size that fits in A4 page with margins) but yo
- (java expects a RNumeric)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `revertTags()`**:** Revert tagged text and images.

Remove all tagged text and images inserted by roogledocs and returns the bare document the tags in place. This does not affect figures and tables inserted by index (i.e. without tags) This is needed if content is being moved around as cut and paste of tagged content unfortunately removes the internal named range of the tag.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$revertTags()
```

Arguments:

- none

Returns: R6 RoogleDocs object: itself - a fluent method

Method `removeTags()`**:** Remove all tags

Finds tags defined in the current document and removes them. This cannot be undone, except by rolling back to a previous version.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$removeTags(confirm)
```

Arguments:

confirm - This action must be confirmed by passing 'true' as cannot be undone. - (defaulting to '(menu(c('Yes','N
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `updateTable()`**:** Update or insert a formatted table into the document.

This function counts the number of tables from the start of the document. Inserting tables by index works only if your document does not change much or you are creating one from scratch. You can overwrite tables using this function but if the order of tables has been changed by your collaborators this will be generally cause probelms. Use of this function is generally discouraged and we now prefer 'updateTaggedTable()'.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateTable(longFormatTable, tableIndex, colWidths, tableWidthInches)
```

Arguments:

longFormatTable A dataframe consisting of the table content and formatting indexed by row and column. at a mi
- (java expects a RDataframe)

tableIndex what is the table index in the document? This can be left out for a new table at the end of the documen
- (java expects a RInteger)

colWidths A vector including the relative length of each column. This can be left out if longFormatTable comes fr
- (java expects a RNumericVector)

tableWidthInches The final width of the table in inches (defaults to a size that fits in A4 page with margins) - (defa
- (java expects a RNumeric)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `updateFigure()`**:** Update or insert a figure in the document from a locally stored PNG by index.

This function counts the number of inline images (i.e. not absolutely positioned ones) from the start of the document. Inserting images by index works only if your document does not change much or you are creating one from scratch. You can overwrite images using this function but if the order of images has been changed by your collaborators this will generally cause problems. This function uploads the image into a temporary file onto your Google Drive, and makes it briefly publicly readable. From there inserts it into the google document. Once this is complete the temporary google drive copy of the image is deleted.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateFigure(absoluteFilePath, figureIndex, dpi, keepUpload)
```

Arguments:

absoluteFilePath a file path to an png image file (only png is supported at this point). - (java expects a RCharacter)

figureIndex what is the figure index in the document? (This only counts inline images - and ignores absolutely pos
- (java expects a RInteger)

dpi the dots per inch of the image in the document (defaults to 300). the final size of the image in the doc will be d
- (java expects a RNumeric)

keepUpload keep the uploaded image as a supplementary file in the same directory as the google doc - (defaulting
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `saveAsPdf()`**:** Save the document as a PDF

Saves a snapshot of the current google doc with 'roogledocs' links removed as a pdf to a local drive. This is mainly intended for snapshotting the current state of the document. For final export once all analysis is complete it may be preferable to call 'doc$removeTags()' and manually export the output but after this no further updating is possible.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$saveAsPdf(absoluteFilePath, uploadCopy)
```

Arguments:

absoluteFilePath - a file path to save the pdf. - (java expects a RFile)

uploadCopy place a copy of the downloaded pdf back onto google drive in the same folder as the document for ex - (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `makeCopy()`**:** Make a copy of the current document

This makes a exact copy of the document under a new name. This name can already exist as googledocs can have multiple files with the same file name but this will certainly lead to confusion later. It is up to the user to create a naming strategy that does not cause issues.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$makeCopy(newName)
```

Arguments:

newName - The new document name. - (java expects a RCharacter)

Returns: R6 RoogleDocs object: a ‘roogledocs‘ object pointing to the new document.

Method `delete()`**:** Delete the current document

Deleted documents can still be retrieved via the Google Drive website but this is otherwise a final operation. After this any operations on this document will fail with a null pointer exception.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$delete(areYouSure)
```

Arguments:

areYouSure - confirm the delete - (defaulting to ‘utils::askYesNo('Are you sure you wan...‘) - (java expects a RLogical)

Returns: void:

Method `uploadSupplementaryFiles()`**:** Upload a file into the same directory as the document.

This allow you to load e.g. a supplementary file, or the pdf of an image file or a docx/html version of a table into google drive into the same directory as the document you are editing. This is handy for organising all the files for a journal submission in one place. Any kind of file can be loaded, and the mimetype will be detected. Normal Google Drive rules for uploads will be triggered at this point. As google drive can have multiple files with the same name the behaviour if the file already exists is slightly complex, with ‘overwrite‘ and ‘duplicate‘ options.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)
```

Arguments:

absoluteFilePath - a file path to upload. - (java expects a RFile)

overwrite - if matching file(s) are found in the target, delete them before uploading the new one. - (defaulting to 'I
- (java expects a RLogical)

duplicate - if matching file(s) are found in the target, upload this new file anyway, creating duplicate names in the
- (java expects a RLogical)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `appendText()`**:** Append text to the document with optional paragraph styling. If you run text blocks into each other without newlines the whole resulting paragraph will be styled. You would normally not want this so it is up to you to end paragraphs with a new line character, before changing styles.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$appendText(text, style)
```

Arguments:

text - a single string with the text to append which may include newlines - (java expects a RCharacter)

style - one of NORMAL_TEXT, TITLE, SUBTITLE, HEADING_1, ... HEADING_6 - (defaulting to ''NORMAL_
- (java expects a RCharacter)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `appendFormattedParagraph()`**:** Append a new paragraph, with text from the 'label' column with optional formating in the other columns.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$appendFormattedParagraph(formattedTextDf)
```

Arguments:

formattedTextDf - a data frame containing the columns label, and optionally: link (as a URL), fontName, fontFac
- (java expects a RDataframe)

Returns: R6 RoogleDocs object: itself - a fluent method

Method `updateCitations()`**:** Update citation tags in the document.
A citation tag is like this '{{cite:challen2020;danon2021}}'. The ids are matched against the provided bibtex, and the tags are replaced with an appropriate citation string. The bibliography

itself is added to a specific slide for references which can be decided with the '{{references}}' tag.
If references do not already exist and there if no '{{references}}' tag the references will be added to the end of the document. Where it can be cut and pasted to the right place. N.B. Do not split up the references when you move them.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleDocs$new(tokenDirectory, disabled);
instance$updateCitations(bibTexPath, citationStyle)
```

Arguments:

bibTexPath - the full file path to the file containing the bibtex - (java expects a RCharacter)

citationStyle - the CSL specification - (defaulting to ''ieee-with-url'') - (java expects a RCharacter)

Returns: R6 RoogleDocs object: itself - a fluent method

Examples

```
## ------------------------------------
## Construct new instance of RoogleDocs
## ------------------------------------
## Not run:
J = roogledocs::JavaApi$get()
# appropriate parameter values must be provided
instance = J$RoogleDocs$new(tokenDirectory, disabled)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$enable()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$enable()

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$disable()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$disable()

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$getName(suffix)`
## ------------------------------------
## Not run:
```

```
# appropriate parameter values must be provided
instance$getName(suffix)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$tagsDefined()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$tagsDefined()

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateTaggedText(text, tagName)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedText(text, tagName)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateTaggedImage(absoluteFilePath, tagName, dpi, keepUpload)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedImage(absoluteFilePath, tagName, dpi, keepUpload)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateTaggedTable(longFormatTable, tagName, colWidths, tableWidthInches)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedTable(longFormatTable, tagName, colWidths, tableWidthInches)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$revertTags()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$revertTags()

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$removeTags(confirm)`
## ------------------------------------
```

```
## Not run:
# appropriate parameter values must be provided
instance$removeTags(confirm)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateTable(longFormatTable, tableIndex, colWidths, tableWidthInches)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTable(longFormatTable, tableIndex, colWidths, tableWidthInches)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateFigure(absoluteFilePath, figureIndex, dpi, keepUpload)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateFigure(absoluteFilePath, figureIndex, dpi, keepUpload)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$saveAsPdf(absoluteFilePath, uploadCopy)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$saveAsPdf(absoluteFilePath, uploadCopy)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$makeCopy(newName)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$makeCopy(newName)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$delete(areYouSure)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$delete(areYouSure)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)`
```

```
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$appendText(text, style)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$appendText(text, style)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$appendFormattedParagraph(formattedTextDf)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$appendFormattedParagraph(formattedTextDf)

## End(Not run)

## ------------------------------------
## Method `RoogleDocs$updateCitations(bibTexPath, citationStyle)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateCitations(bibTexPath, citationStyle)

## End(Not run)
```

RoogleSlides *RoogleSlides*

Description

Programmatically substitute images, data and tables into a google presentation.

Version: 0.5.0

Generated: 2024-04-27T13:56:10.641295657

Arguments

tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

`disabled` disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Details

The purpose being to support google slides as a platform for interactive development and documentation of data analysis in R. The workflow supported is a parallel documentation and analysis where a team of people are working collaboratively on documentation, whilst at the same time analysis is being performed and results updated repeatedly as a result of new data. In this environment updating numeric results, tabular data and figures in word documents manually becomes annoying. With roogledocs you can automate this a bit like a RMarkdown document, but with the added benefit that the content can be updated independently of the analysis, by the wider team.

Methods

Constructors:

- `J$RoogleSlides$new(tokenDirectory, disabled)`

Class methods:

- `instance$enable()`
- `instance$disable()`
- `instance$getName(suffix)`
- `instance$tagsDefined()`
- `instance$updateTaggedText(text, tagName)`
- `instance$updateTaggedImage(absoluteFilePath, tagName, keepUpload)`
- `instance$updateTaggedTable(longFormatTable, tagName, colWidths)`
- `instance$removeTags(confirm)`
- `instance$saveAsPdf(absoluteFilePath, uploadCopy)`
- `instance$makeCopy(newName)`
- `instance$delete(areYouSure)`
- `instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)`
- `instance$appendFormattedSlide(title, formattedTextDf)`
- `instance$slideDimensions()`
- `instance$slideLayouts()`
- `instance$setDefaultLayout(layout)`
- `instance$updateCitations(bibTexPath, citationStyle)`
- `instance$clone()`
- `instance$print()`

Constructor `new()`**:** Create a RoogleSlides object for managing the interaction.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
```

Arguments:

tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaul
- (java expects a RCharacter)

disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can b
- (java expects a RLogical)

Returns: the new R6 RoogleSlides object

Method `enable()`**:** Enables roogledocs method calls for this document.
It is likely one of 'withDocument()', 'findOrCreateDocument()' or 'findOrCloneTemplate()' methods will be needed to specify the document.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$enable()
```

Arguments:

- none

Returns: R6 RoogleSlides object: itself - a fluent method

Method `disable()`**:** Disables roogledocs temporarily for this document.
While disabled all calls to roogledocs will silently abort.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$disable()
```

Arguments:

- none

Returns: R6 RoogleSlides object: itself - a fluent method

Method `getName()`**:** Return the name of the presentation

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$getName(suffix)
```

Arguments:

suffix an additional suffix to add to the name - (defaulting to "") - (java expects a RCharacter)

Returns: String:

Method `tagsDefined()`**:** List all tags
Finds tags defined in the current document

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$tagsDefined()
```

Arguments:

- none

Returns: RDataframe: a dataframe containing tag and count columns

Method `updateTaggedText()`**:** Replace tags for text

Substitutes all occurrences of {{tag-name}} with the text parameter. If the tag is not found then a new slide is inserted at the end in a section titled "Unmatched tags:". From there they can be cut and pasted into the right place.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$updateTaggedText(text, tagName)
```

Arguments:

text the value to replace the tag with (e.g. a result from analysis) (cannot be empty) - (java expects a RCharacter)

tagName the tag name - (defaulting to 'deparse(substitute(text))') - (java expects a RCharacter)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `updateTaggedImage()`**:** Replace a tag with an image.

Substitutes all occurrences of {{tag-name}} with an image from the local storage.

The image is uploaded to your google drive as a temporary file, and made publicly readable. From there it is inserted into the google slides, and once completed the temporary file deleted from your google drive, unless 'keepUpload' is true. Insertion is done in the dimensions of the containing box of the image if it already exists or a default slide body box if not.

If the tag is not found in the document a new slide will be created at the end of the presentation with the image and an uninformative title which can be changed.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$updateTaggedImage(absoluteFilePath, tagName, keepUpload)
```

Arguments:

absoluteFilePath a file path to an png image file. - (java expects a RCharacter)

tagName the tag name - (defaulting to 'deparse(substitute(absoluteFilePath))') - (java expects a RCharacter)

keepUpload keep the uploaded image as a supplementary file in the same directory as the google doc. N.B. the res - (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `updateTaggedTable()`**:** Replace a tag with a table.

Substitutes a unique occurrence of {{tag-name}} with a table. The tag must either be in a text box shape or as the first entry in a table. Once inserted the table is tagged using a zero width character as the very first item in the first cell. This will be removed if 'removeTags()' is called.

If the tag is not found in the document a new slide will be created at the end with the table.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$updateTaggedTable(longFormatTable, tagName, colWidths)
```

Arguments:

longFormatTable A dataframe consisting of the table content and formatting indexed by row and column. at a mi - (java expects a RDataframe)

tagName the tag name - (defaulting to 'deparse(substitute(longFormatTable))') - (java expects a RCharacter)

colWidths A vector including the relative length of each column. This can be left out if longFormatTable comes fr - (java expects a RNumericVector)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `removeTags()`**:** Remove all tags

Finds tags defined in the current document and removes them. This cannot be undone, except by rolling back to a previous version.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$removeTags(confirm)
```

Arguments:

confirm - This action must be confirmed by passing 'true' as cannot be undone. - (defaulting to '(menu(c('Yes','N - (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `saveAsPdf()`**:** Save the document as a PDF

Saves a snapshot of the current google slides with 'roogledocs' links removed as a pdf to a local drive. This is mainly intended for snap-shotting the current state of the document. For final export once all analysis is complete it may be preferable to call 'doc$removeTags()' and manually export the output but after this no further updating is possible.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$saveAsPdf(absoluteFilePath, uploadCopy)
```

Arguments:

absoluteFilePath - a file path to save the pdf. - (java expects a RFile)

uploadCopy place a copy of the downloaded pdf back onto google drive in the same folder as the document for ex - (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `makeCopy()`**:** Make a copy of the current document

This makes a exact copy of the document under a new name. This name can already exist as googledocs can have multiple files with the same file name but this will certainly lead to confusion later. It is up to the user to create a naming strategy that does not cause issues.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$makeCopy(newName)
```

Arguments:

newName - The new document name. - (java expects a RCharacter)

Returns: R6 RoogleSlides object: a ‘roogledocs‘ object pointing to the new document.

Method `delete()`**:** Delete the current presentation

Deleted presentations can still be retrieved via the Google Drive website but this is otherwise a final operation. After this any operations on this presentation will fail with a null pointer exception.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$delete(areYouSure)
```

Arguments:

areYouSure - confirm the delete - (defaulting to ‘utils::askYesNo('Are you sure you wan...‘) - (java expects a RLogical)

Returns: void:

Method `uploadSupplementaryFiles()`**:** Upload a file into the same directory as the document.

This allow you to load e.g. a supplementary file, or the pdf of an image file or a docx/html version of a table into google drive into the same directory as the slides you are editing. This is handy for organising all the files for a journal submission in one place. Any kind of file can be loaded, and the mimetype will be detected. Normal Google Drive rules for uploads will be triggered at this point. As google drive can have multiple files with the same name the behaviour if the file already exists is slightly complex, with ‘overwrite‘ and ‘duplicate‘ options.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)
```

Arguments:

absoluteFilePath - a file path to upload. - (java expects a RFile)

overwrite - if matching file(s) are found in the target, delete them before uploading the new one. - (defaulting to 'I
- (java expects a RLogical)

duplicate - if matching file(s) are found in the target, upload this new file anyway, creating duplicate names in the
- (java expects a RLogical)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `appendFormattedSlide()`**:** Append a new "TITLE_AND_BODY" slide, with formatted text from the 'label' column with optional formating in the other columns.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$appendFormattedSlide(title, formattedTextDf)
```

Arguments:

title - A plain text title - (java expects a RCharacter)

formattedTextDf - a data frame containing the columns label, and optionally: link (as a URL), fontName, fontFac
- (java expects a RDataframe)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `slideDimensions()`**:** The default dimensions of the body of a new slide.

A new slide will be created using a default layout which is usually the slide layout with the biggest text box on it. This can be set manually with the 'setDefaultLayout()' function.

The body is the largest text box on the slide. This is the place that images or tables will be placed by default. If you want to fit in with the theme then images will need to be to be sized to these dimensions to fill the slide.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$slideDimensions()
```

Arguments:

- none

Returns: RNamedList: a named list with width and height in inches, and the name of the layout being used as the default.

Method `slideLayouts()`**:** The layouts available in the slides templates

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$slideLayouts()
```

Arguments:

- none

Returns: RCharacterVector: a list of available slide layouts

Method `setDefaultLayout()`**:** Set the default layout

The default layout for the slide is used when inserting new slides at the end of the document for images. A default layout will have 2 text boxes, one for the title and one for the content. The second text box will be large. The layouts in a presentation can be listed with ‘slideLayouts()‘ or seen on the google slides ‘Apply Layout...‘ menu option.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$setDefaultLayout(layout)
```

Arguments:

layout a string representing the layout - (java expects a RCharacter)

Returns: R6 RoogleSlides object: itself - a fluent method

Method `updateCitations()`**:** Update citation tags in the document.

A citation tag is like this ‘{{cite:challen2020;danon2021}}‘. The ids are matched against the provided bibtex, and the tags are replaced with an appropriate citation string. The bibliography itself is added to a specific slide for references which can be decided with the ‘{{references}}‘ tag.

If references do not already exist and there if no ‘{{references}}‘ tag a new slide will be created at the end of the presentation.

Usage:

```
J = roogledocs::JavaApi$get()
instance = J$RoogleSlides$new(tokenDirectory, disabled);
instance$updateCitations(bibTexPath, citationStyle)
```

Arguments:

bibTexPath - the full file path to the file containing the bibtex - (java expects a RCharacter)

citationStyle - the CSL specification - (defaulting to ‘’ieee’‘) - (java expects a RCharacter)

Returns: R6 RoogleSlides object: itself - a fluent method

Examples

```
## ------------------------------------
## Construct new instance of RoogleSlides
## ------------------------------------
## Not run:
J = roogledocs::JavaApi$get()
# appropriate parameter values must be provided
instance = J$RoogleSlides$new(tokenDirectory, disabled)

## End(Not run)
```

```
## ------------------------------------
## Method `RoogleSlides$enable()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$enable()

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$disable()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$disable()

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$getName(suffix)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$getName(suffix)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$tagsDefined()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$tagsDefined()

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$updateTaggedText(text, tagName)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedText(text, tagName)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$updateTaggedImage(absoluteFilePath, tagName, keepUpload)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedImage(absoluteFilePath, tagName, keepUpload)

## End(Not run)
```

```
## ------------------------------------
## Method `RoogleSlides$updateTaggedTable(longFormatTable, tagName, colWidths)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateTaggedTable(longFormatTable, tagName, colWidths)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$removeTags(confirm)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$removeTags(confirm)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$saveAsPdf(absoluteFilePath, uploadCopy)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$saveAsPdf(absoluteFilePath, uploadCopy)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$makeCopy(newName)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$makeCopy(newName)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$delete(areYouSure)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$delete(areYouSure)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$uploadSupplementaryFiles(absoluteFilePath, overwrite, duplicate)
```

```
## End(Not run)

## ------------------------------------
## Method `RoogleSlides$appendFormattedSlide(title, formattedTextDf)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$appendFormattedSlide(title, formattedTextDf)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$slideDimensions()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$slideDimensions()

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$slideLayouts()`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$slideLayouts()

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$setDefaultLayout(layout)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$setDefaultLayout(layout)

## End(Not run)

## ------------------------------------
## Method `RoogleSlides$updateCitations(bibTexPath, citationStyle)`
## ------------------------------------
## Not run:
# appropriate parameter values must be provided
instance$updateCitations(bibTexPath, citationStyle)

## End(Not run)
```

`search_for_documents` *Search for documents with the given title*

Description

no description

Usage

```
search_for_documents(
 titleMatch,
 tokenDirectory
)
```

Arguments

titleMatch	titleMatch a string to be searched for as an approximate match. All results will be retrieved with document ids. - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

Value

RDataframe: a dataframe containing id and name columns

---

search_for_slides *Search for documents with the given title*

Description

no description

Usage

```
search_for_slides(
 titleMatch,
 tokenDirectory
)
```

Arguments

titleMatch	titleMatch a string to be searched for as an approximate match. All results will be retrieved with document ids. - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)

Value

RDataframe: a dataframe containing id and name columns

slides_by_id *Get a document by id or sharing link.*

Description

no description

Usage

```
slides_by_id(
 shareUrlOrDocId,
 tokenDirectory,
 disabled
)
```

Arguments

shareUrlOrDocId	shareUrlOrDocId the url from clicking a share button in google slides or an id from searchForDocuments() method - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleSlides object: itself - a fluent method

slides_by_name *Get a document by name or create a blank document if missing.*

Description

no description

Usage

```
slides_by_name(
 title,
 tokenDirectory,
 disabled
)
```

Arguments

title	title a document title. If there is an exact match in google drive then that document will be used - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleSlides object: itself - a fluent method

---

slides_from_template *Get a document by name or create one from a template if missing.*

Description

no description

Usage

```
slides_from_template(
 title,
 templateUri,
 tokenDirectory,
 disabled
)
```

Arguments

title	title a document title. If there is an exact match in google drive then that document will be used otherwise a new one will be created. - (java expects a RCharacter)
templateUri	templateUri the share link (or document id) of a template google document - (java expects a RCharacter)
tokenDirectory	tokenDirectory the place to store authentication tokens. This should not be checked into version control. - (defaulting to '.tokenDirectory()') - (java expects a RCharacter)
disabled	disabled a flag to switch roogledocs off (on a document by document basis, for testing or development. This can be set globally with 'options('roogledocs.disabled'=TRUE)' - (defaulting to 'getOption('roogledocs.disabled',FALSE)') - (java expects a RLogical)

Value

R6 RoogleSlides object: itself - a fluent method

Index