

Package ‘easyreporting’

November 19, 2024

Type Package

Title Helps creating report for improving Reproducible Computational Research

Version 1.19.0

Description An S4 class for facilitating the automated creation of markdown files inside other packages/software even without knowing markdown language. Best if implemented in functions as ``recursive" style programming.

License Artistic-2.0

Encoding UTF-8

biocViews ReportWriting

LazyData false

BugReports <https://github.com/drighelli/easyreporting/issues>

Imports rmarkdown, methods, tools, shiny, rlang

Suggests distill, BiocStyle, knitr, readxl, edgeR, limma, EDASeq, statmod

RoxygenNote 7.1.1

VignetteBuilder knitr

git_url <https://git.bioconductor.org/packages/easyreporting>

git_branch devel

git_last_commit 0f50bcf

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-19

Author Dario Righelli [cre, aut]

Maintainer Dario Righelli <dario.righelli@gmail.com>

Contents

addResource	3
addResource,easyreporting-method	3
compile	4
compile,easyreporting-method	5
easyreporting	5
erGUIVolcano	7
getBibliography	7
getBibliography,easyreporting-method	8
getOptionsList	8
getOptionsList,easyreporting-method	9
getReportFilename	10
getReportFilename,easyreporting-method	10
initReportFilename	11
initReportFilename,easyreporting-method	12
makeOptionsList	13
mkdCodeChunkCommented	14
mkdCodeChunkCommented,easyreporting-method	15
mkdCodeChunkComplete	16
mkdCodeChunkComplete,easyreporting-method	17
mkdCodeChunkEnd	18
mkdCodeChunkEnd,easyreporting-method	18
mkdCodeChunkSt	19
mkdCodeChunkSt,easyreporting-method	20
mkdCodeChunkTitledCommented	21
mkdCodeChunkTitledCommented,easyreporting-method	22
mkdGeneralMsg	23
mkdGeneralMsg,easyreporting-method	24
mkdGeneralTitledMsg	24
mkdGeneralTitledMsg,easyreporting-method	25
mkdSetGlobalOpts	26
mkdSetGlobalOpts,easyreporting-method	26
mkdSourceFiles	27
mkdSourceFiles,easyreporting-method	27
mkdTitle	28
mkdTitle,easyreporting-method	28
mkdVariableAssignment	29
mkdVariableAssignment,easyreporting-method	30
setBibliography	31
setBibliography,easyreporting-method	32
setOptionsList	32
setOptionsList,easyreporting-method	34

addResource	<i>addResource</i>
-------------	--------------------

Description

addResource

Usage

```
addResource(object, source, reference, description)
```

Arguments

object	an easyreporting class instance
source	a string indicating the reference (i.e. "GEO")
reference	a string indicanting the reference of the source (i.e. "GSE60231")
description	a natural language description

Value

an easyreporting class instance

Examples

```
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
rd <- addResource(rd, source="GEO", "GSE60231", "Transcriptome of BMDC to different antigen delivery systems")
## Not run:
compile(rd)

## End(Not run)
```

addResource, easyreporting-method	<i>addResource</i>
-----------------------------------	--------------------

Description

addResource

Usage

```
## S4 method for signature 'easyreporting'
addResource(object, source, reference, description)
```

Arguments

object	an easyreporting class instance
source	a string indicating the reference (i.e. "GEO")
reference	a string indicanting the reference of the source (i.e. "GSE60231")
description	a natural language description

Value

an easyreporting class instance

Examples

```
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
rd <- addResource(rd, source="GEO", "GSE60231", "Transcriptome of BMDC to different antigen delivery systems")
## Not run:
compile(rd)

## End(Not run)
```

compile

compile

Description

prints the sessionInfo and compiles the produced rmarkdown file into an HTML report.

Usage

```
compile(object)
```

Arguments

object	an easyreporting class object
--------	-------------------------------

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report", title="example_report",
                    author=c("It's me"))
compile(rd)

## End(Not run)
```

compile,easyreporting-method
compile

Description

prints the sessionInfo and compiles the rmarkdown file

Usage

```
## S4 method for signature 'easyreporting'  
compile(object)
```

Arguments

object an easyreporting class object

Value

none

Examples

```
## Not run:  
rd <- easyreporting(filenamePath="./project_report", title="example_report",  
                    author=c("It's me"))  
  
compile(rd)  
  
## End(Not run)
```

easyreporting *easyreporting*

Description

An S4 class for managing rmarkdown report. Each instance describes an rmarkdown file.

Usage

```
easyreporting(  
  filenamePath,  
  title,  
  author,  
  optionList = NULL,  
  documentType = "rmarkdown::html_document",  
  bibfile = ""
```

```

)

easyreporting(
  filenamePath,
  title,
  author,
  optionList = NULL,
  documentType = "rmarkdown::html_document",
  bibfile = ""
)

```

Arguments

filenamePath	the path with the name of the rmarkdown.
title	the title of the report section.
author	the author(s) of the report.
optionList	a list of options for the general rmarkdown document.
documentType	type of report final document, if author(s) is a person it will be <code>distill::distill_article</code> (<code>rmarkdown::html_document</code> default)
bibfile	a bibfile for bibliography.

Value

an S4 `easyreporting` class instance

Slots

filenamePath	the path with the name of the rmarkdown.
title	the title of the report section.
author	the author(s) of the report.
type of report final document,	if author(s) is a person it will be <code>distill::distill_article</code> (<code>rmarkdown::html_document</code> default)
bibfile	a bibfile for bibliography.
optionList	a list of options for the general rmarkdown document.

Examples

```

rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author="It's me")

```

erGUIVolcano	<i>erGUIVolcano</i>
--------------	---------------------

Description

executes a Shiny GUI to perform a volcano plot and trace its executed functions.

Usage

```
erGUIVolcano()
```

Value

none

Examples

```
## Not run:  
erGUIVolcano()  
  
## End(Not run)
```

getBibliography	<i>getBibliography</i>
-----------------	------------------------

Description

returns the bibfile name attached to the object

Usage

```
getBibliography(object)
```

Arguments

object an easyreporting class object

Value

none

Examples

```
## Not run:  
TBD  
  
## End(Not run)
```

getBibliography, easyreporting-method
getBibliography

Description

returns the bibfile name attached to the object

Usage

```
## S4 method for signature 'easyreporting'  
getBibliography(object)
```

Arguments

object an easyreporting class object

Value

none

Examples

```
example(easyreporting)  
getBibliography(rd)
```

getOptionsList *getOptionsList*

Description

returns the optionList from the easyreporting class (see the makeOptionList function for more details).

Usage

```
getOptionsList(object)
```

Arguments

object an easyreporting class object

Value

a list of options

Examples

```
## Not run:  
rd <- easyreporting(filenamePath="./project_report", title="example_report",  
                    author=c("It's me"))  
optList <- getOptionsList(rd)  
  
## End(Not run)
```

getOptionsList,easyreporting-method
getOptionsList

Description

returns the optionList from the easyreporting class

Usage

```
## S4 method for signature 'easyreporting'  
getOptionsList(object)
```

Arguments

object an easyreporting class object

Value

a list of options

Examples

```
## Not run:  
rd <- easyreporting(filenamePath="./project_report", title="example_report",  
                    author=c("It's me"))  
optList <- getOptionsList(rd)  
  
## End(Not run)
```

`getReportFilename` *getReportFilename*

Description

returns the filename with its path of the report generated by easyreporting

Usage

```
getReportFilename(object)
```

Arguments

`object` an easyreporting class object

Value

a string of report file name with path

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
(rep <- getReportFilename(rd))

## End(Not run)
```

`getReportFilename,easyreporting-method`
getReportFilename

Description

returns the report filename with path

Usage

```
## S4 method for signature 'easyreporting'
getReportFilename(object)
```

Arguments

`object` an easyreporting class object

Value

a string of report file name with path

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
(rep <- getReportFilename(rd))

## End(Not run)
```

`initReportFilename` *initReportFilename*

Description

`initReportFilename`

Usage

```
initReportFilename(
  object,
  filepath = NULL,
  title = NULL,
  author = NULL,
  optionList = NULL,
  bibfile = NULL,
  documentType = NULL,
  resources = NULL
)
```

Arguments

<code>object</code>	an easyreporting class object
<code>filepath</code>	the name of the report with or without the path
<code>author</code>	the name of the report author
<code>documentType</code>	type of report final document. (html is default)
<code>mainTitle</code>	the title of the report

Value

an easyreporting class object

`initReportFilename,easyreporting-method`
initReportFilename

Description

internal funtion usefull to init the report for the first time.

Usage

```
## S4 method for signature 'easyreporting'  
initReportFilename(  
  object,  
  filepath = NULL,  
  title = NULL,  
  author = NULL,  
  optionList = NULL,  
  bibfile = "",  
  documentType = "rmarkdown::html_document"  
)
```

Arguments

<code>object</code>	an easyreporting class object
<code>filepath</code>	the name of the report with or without the path
<code>author</code>	the name of the report author as a person format (where expected comment named fields are ORCID, url, affiliation, affiliation_url). In case author is a character, it will be reported as it is.
<code>bibfile</code>	a bibfile for bibliography.
<code>documentType</code>	type of report final document, if author(s) is a person it will be <code>distill::distill_article</code> (<code>rmarkdown::html_document</code> default)
<code>mainTitle</code>	the title of the report

Value

an easyreporting class object

See Also

[person](#)

makeOptionsList *makeOptionsList*

Description

makes an list of rmarkdown options

Usage

```
makeOptionsList(  
  cacheFlag = TRUE,  
  evalFlag = TRUE,  
  echoFlag = TRUE,  
  warningFlag = FALSE,  
  showMessages = FALSE,  
  includeFlag = TRUE,  
  collapseFlag = FALSE,  
  purlFlag = TRUE,  
  errorFlag = TRUE,  
  messageFlag = TRUE,  
  highlightFlag = TRUE,  
  promptFlag = FALSE,  
  stripWhiteFlag = TRUE,  
  tidyFlag = FALSE  
)
```

Arguments

cacheFlag	boolean for caching chunk data (default TRUE)
evalFlag	boolean for evaluating the code chunk in the compiled version (default TRUE)
echoFlag	boolean for showing the code chunk (default TRUE)
warningFlag	boolean for showing the chunk warnings (default FALSE)
showMessages	boolean for showing the chunk warnings in compiled version (default FALSE)
includeFlag	boolean for including the code chunk in the compiled version (default TRUE)
collapseFlag	boolean for collapsing the code chunk in the compiled version (default FALSE),
purlFlag	boolean for extracting the code chunk as R code in a separate R file (default TRUE),
errorFlag	boolean for including the error generated by the code chunk in the compiled version (default TRUE),
messageFlag	boolean for including the code chunk messages in the compiled version (default TRUE),
highlightFlag	boolean for highlighting the code chunk in the compiled version (default TRUE),
promptFlag	boolean for including a ">" for the code chunk in the compiled version (default FALSE),

`stripWhiteFlag` boolean for removing the white spaces at beginning/end of the code chunk in the compiled version (default TRUE),

`tidyFlag` boolean for creating a tidy code chunk in the compiled version (default FALSE).

Value

list of rmarkdown options

Examples

```
optList <- makeOptionsList()
```

`mkdCodeChunkCommented` *mkdCodeChunkCommented*

Description

it creates a complete code chunk, adding a natural language comment before of it.

Usage

```
mkdCodeChunkCommented(
  object,
  comment = NULL,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

`object` an easyreporting class object

`comment` a string with the natural language comment for the chunk.

`code` a string within the code.

`optionList` a list of options (default is the class options).

`sourceFilesList` a optional list of files to source inside the chunk.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
optList <- makeOptionsList(includeFlag=TRUE, cacheFlag=TRUE)
mkdCodeChunkCommented(rd,
                      comment="This is the comment of the following code chunk",
                      code="a <- 1\n b <- 2\n (c <- a+b)\n", optionList=optList,
                      sourceFilesList=NULL)

## End(Not run)
```

```
mkdCodeChunkCommented,easyreporting-method
      mkdCodeChunkCommented
```

Description

it creates a complete code chunk, adding a natural language comment before of it.

Usage

```
## S4 method for signature 'easyreporting'
mkdCodeChunkCommented(
  object,
  comment = NULL,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

object	an easyreporting class object
comment	a string with the natural language comment for the chunk.
code	a string within the code.
optionList	a list of options (default is the class options).
sourceFilesList	a optional list of files to source inside the chunk.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
optList <- makeOptionsList(includeFlag=TRUE, cacheFlag=TRUE)
mkdCodeChunkCommented(rd,
                      comment="This is the comment of the following code chunk",
                      code=c(quote(a <- 1), quote(b <- 2), quote(c <- a+b)),
                      optionList=optList, sourceFilesList=NULL)
mkdCodeChunkCommented(rd,
                      comment="This is the comment of the following code chunk",
                      code="a <- 1\n b <- 2\n(c <- a+b)\n", optionList=optList,
                      sourceFilesList=NULL)

## End(Not run)
```

mkdCodeChunkComplete *mkdCodeChunkComplete*

Description

it creates a complete code chunk.

Usage

```
mkdCodeChunkComplete(
  object,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

object	an easyreporting class object
code	a string or an expression generated with <code>quote</code> containing a function call or the entire code chunk to trace.
optionList	a list of options.
sourceFilesList	a list of files to source.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdCodeChunkComplete(rd, code="a <- 1\n b <- 2\n c <- a+b\n print(c)")

## End(Not run)
```

mkdCodeChunkComplete,easyreporting-method
mkdCodeChunkComplete

Description

it creates a complete code chunk.

Usage

```
## S4 method for signature 'easyreporting'
mkdCodeChunkComplete(
  object,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

object	an easyreporting class object
code	a string or an expression (or a list of expressions) generated with quote containing a function call or the entire code chunk to trace.
optionList	a list of options.
sourceFilesList	a list of files to source.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdCodeChunkComplete(rd, code=c(quote(a <- 1), quote(b <- 2), quote(c<- a+b),
quote(print(c))))
mkdCodeChunkComplete(rd, code="a <- 1\n b <- 2\n c <- a+b\n print(c)")

## End(Not run)
```

mkdCodeChunkEnd	<i>mkdCodeChunkEnd</i>
-----------------	------------------------

Description

it creates a code chunk end. Always use it after a `mkdCodeChunkSt()`

Usage

```
mkdCodeChunkEnd(object)
```

Arguments

object an easyreporting class object

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdCodeChunkSt(rd)
## just leaving empty
mkdCodeChunkEnd(rd)

## End(Not run)
```

mkdCodeChunkEnd,easyreporting-method	<i>mkdCodeChunkEnd</i>
--------------------------------------	------------------------

Description

it creates a code chunk end. Always use it after a `mkdCodeChunkSt()`

Usage

```
## S4 method for signature 'easyreporting'
mkdCodeChunkEnd(object)
```

Arguments

object an easyreporting class object

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

mkdCodeChunkSt(rd)
## just leaving empty
mkdCodeChunkEnd(rd)

## End(Not run)
```

mkdCodeChunkSt	<i>mkdCodeChunkSt</i>
----------------	-----------------------

Description

it creates a code chunk start. A list of options and files to source for the chunk can optionally be passed to the function.

Usage

```
mkdCodeChunkSt(
  object,
  optionList = getOptionsList(object),
  sourceFilesList = NULL,
  isComplete = FALSE
)
```

Arguments

object	an easyreporting class object
optionList	a list of options
sourceFilesList	a list of files that can be sourced inside the code chunk.
isComplete	a flag determining if the chunk is already a complete chunk

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

## no options
mkdCodeChunkSt(rd)
## just leaving empty
mkdCodeChunkEnd(rd)

## setting options
optList <- makeOptionsList(includeFlag=TRUE)
mkdCodeChunkSt(rd, optionList=optList)
## just leaving empty
mkdCodeChunkEnd(rd)

## End(Not run)
```

```
mkdCodeChunkSt,easyreporting-method
      mkdCodeChunkSt
```

Description

it creates a code chunk start. A list of options and files to source for the chunk can optionally be passed to the function.

Usage

```
## S4 method for signature 'easyreporting'
mkdCodeChunkSt(
  object,
  optionList = getOptionsList(object),
  sourceFilesList = NULL,
  isComplete = FALSE
)
```

Arguments

object	an easyreporting class object
optionList	a list of options
sourceFilesList	a list of files that can be sourced inside the code chunk.
isComplete	a flag determining if the chunk is already a complete chunk

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

## no options
mkdCodeChunkSt(rd)
## just leaving empty
mkdCodeChunkEnd(rd)

## setting options
optList <- makeOptionsList(includeFlag=TRUE)
mkdCodeChunkSt(rd, optionList=optList)
## just leaving empty
mkdCodeChunkEnd(rd)

## End(Not run)
```

```
mkdCodeChunkTitledCommented
      mkdCodeChunkCommented
```

Description

it creates a complete code chunk, adding a natural language comment before of it.

Usage

```
mkdCodeChunkTitledCommented(
  object,
  title = NULL,
  level = 1,
  comment = NULL,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

object	an easyreporting class object
title	the title to assign to the code chunk section
level	the level of the title (default is 1)
comment	a string with the natural language comment for the chunk.
code	a string within the code.
optionList	a list of options (default is the class options).
sourceFilesList	a optional list of files to source inside the chunk.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
optList <- makeOptionsList(includeFlag=TRUE, cacheFlag=TRUE)
mkdCodeChunkCommented(rd,
                      comment="This is the comment of the following code chunk",
                      code="a <- 1\n b <- 2\n (c <- a+b)\n", optionList=optList,
                      sourceFilesList=NULL)

## End(Not run)
```

mkdCodeChunkTitledCommented,easyreporting-method
mkdCodeChunkTitledCommented

Description

it creates a complete code chunk, adding a natural language comment before of it.

Usage

```
## S4 method for signature 'easyreporting'
mkdCodeChunkTitledCommented(
  object,
  title = NULL,
  level = 1,
  comment = NULL,
  code,
  optionList = getOptionsList(object),
  sourceFilesList = NULL
)
```

Arguments

<code>object</code>	an easyreporting class object
<code>title</code>	the title to assign to the code chunk section
<code>level</code>	the level of the title (default is 1)
<code>comment</code>	a string with the natural language comment for the chunk.
<code>code</code>	a string within the code.
<code>optionList</code>	a list of options (default is the class options).
<code>sourceFilesList</code>	a optional list of files to source inside the chunk.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
optList <- makeOptionsList(includeFlag=TRUE, cacheFlag=TRUE)
mkdCodeChunkTitledCommented(rd, title="Title Example", level=1,
                             comment="This is the comment of the following code chunk",
                             code="a <- 1\n b <- 2\n(c <- a+b)\n", optionList=optList,
                             sourceFilesList=NULL)

## End(Not run)
```

mkdGeneralMsg

mkdGeneralMsg

Description

It appends a general message to the report, useful for describing a code chunk or a part of the report. Useful for adding natural language comments.

Usage

```
mkdGeneralMsg(object, message)
```

Arguments

object	an easyreporting class object
message	the message to append to the report

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdGeneralMsg(rd, message="Writing a paragraph to describe my code chunk")

## End(Not run)
```

mkdGeneralMsg, easyreporting-method
mkdGeneralMsg

Description

It appends a general message to the report. Useful for adding natural language comments.

Usage

```
## S4 method for signature 'easyreporting'  
mkdGeneralMsg(object, message)
```

Arguments

object	an easyreporting class object
message	the message to append to the report

Value

none

Examples

```
## Not run:  
rd <- easyreporting(filenamePath="./project_report",  
                    title="example_report", author=c("It's me"))  
mkdGeneralMsg(rd, "Writing a paragraph to describe my code chunk")  
  
## End(Not run)
```

mkdGeneralTitledMsg *mkdGeneralTitledMsg*

Description

It appends a a titled section followed by a general message to the report. Useful for adding natural language comments.

Usage

```
mkdGeneralTitledMsg(object, title = NULL, level = 1, message)
```


Arguments

object	an easyreporting class object
title	the title of the report section.
level	the level (1 to 6) of the title (default is 1)
message	the message to append to the report

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdGeneralTitledMsg(rd, title="Generic SubTitle for this message", level=2,
                    message="Writing a paragraph to describe my code chunk")

## End(Not run)
```

mkdGeneralTitledMsg,easyreporting-method
mkdGeneralTitledMsg

Description

It appends a a titled section followed by a general message to the report. Useful for adding natural language comments.

Usage

```
## S4 method for signature 'easyreporting'
mkdGeneralTitledMsg(object, title = NULL, level = 1, message)
```

Arguments

object	an easyreporting class object
title	the optional title to give to the message section
level	the level (1 to 6) of the title (default is 1)
message	the message to append to the report

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
mkdGeneralTitledMsg(rd, title="Generic SubTitle for this message", level=2,
                    message="Writing a paragraph to describe my code chunk")

## End(Not run)
```

`mkdSetGlobalOpts` *mkdSetGlobalOpts*

Description

`mkdSetGlobalOpts`

Usage

```
mkdSetGlobalOpts(object, optionList = list())
```

Arguments

`object` an easyreporting class object
`optionList` a list of options

Value

an easyreporting class object

`mkdSetGlobalOpts,easyreporting-method`
mkdSetGlobalOpts

Description

internal function for appending to the report the initial options setup

Usage

```
## S4 method for signature 'easyreporting'
mkdSetGlobalOpts(object, optionList = list())
```

Arguments

`object` an easyreporting class object
`optionList` a list of options

Value

an easyreporting class object

mkdSourceFiles	<i>mkdSourceFiles</i>
----------------	-----------------------

Description

includes a list of source files inside the rmarkdown

Usage

```
mkdSourceFiles(object, ...)
```

Arguments

object	an easyreporting class object
...	a list of files to source

Value

none

mkdSourceFiles, easyreporting-method	<i>mkdSourceFiles</i>
--------------------------------------	-----------------------

Description

includes a list of source files inside the rmarkdown

Usage

```
## S4 method for signature 'easyreporting'
mkdSourceFiles(object, ...)
```

Arguments

object	an easyreporting class object
...	a list of files to source with path

Value

none

mkdTitle	<i>mkdTitle</i>
----------	-----------------

Description

Inserts an rmarkdown title inside the report, useful for create distinct sections inside the final respot.

Usage

```
mkdTitle(object, title, level = 1)
```

Arguments

object	an easyreporting class object
title	a string within the title.
level	a numeric from 1 to 6 (default is 1).

Value

none

Examples

```
## Not run:  
rd <- easyreporting(filenamePath="./project_report",  
                    title="example_report", author=c("It's me"))  
  
mkdTitle(rd, "First Level Title")  
mkdTitle(rd, "Sub-Title", level=2)  
  
## End(Not run)
```

mkdTitle,easyreporting-method	<i>mkdTitle</i>
-------------------------------	-----------------

Description

Inserts an rmarkdown title inside the report.

Usage

```
## S4 method for signature 'easyreporting'  
mkdTitle(object, title, level = 1)
```

Arguments

object an easyreporting class object
 title a string within the title.
 level a numeric from 1 to 6 (default is 1).

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

mkdTitle(rd, "First Level Title")
mkdTitle(rd, "Sub-Title", level=2)

## End(Not run)
```

mkdVariableAssignment *mkdVariableAssignment*

Description

it includes a variable assignment in the report. NB: a call to the "mkdCodeChunkSt" has to be done before using it.

Usage

```
mkdVariableAssignment(object, variable.name, variable.object.name, show)
```

Arguments

object an easyreporting class object
 variable.name a string indicating the name of the variabe to store in the report. (This can be changed here, but further uses of the variable needs to take into account the variable name change).
 variable.object.name the name of the already existing variable. (This cannot be canged.)
 show a boolean indicating if to show the message before writing it into the rmarkdown file.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
## leaving the default options to the code chunk
mkdCodeChunkSt(rd)
## adding a variable assignment
variable <- 1
mkdVariableAssignment(rd, "variable", "variable", show=TRUE)
mkdCodeChunkEnd(rd)

## End(Not run)
```

```
mkdVariableAssignment,easyreporting-method
      mkdVariableAssignment
```

Description

it includes a variable assignment in the report. NB: a call to the "mkdCodeChunkSt" has to be done before using it.

Usage

```
## S4 method for signature 'easyreporting'
mkdVariableAssignment(
  object,
  variable.name,
  variable.object.name,
  show = FALSE
)
```

Arguments

<code>object</code>	an easyreporting class object
<code>variable.name</code>	a string indicating the name of the variable to store in the report. (This can be changed here, but further uses of the variable needs to take into account the variable name change).
<code>variable.object.name</code>	the name of the already existing variable. (This cannot be changed.)
<code>show</code>	a boolean indicating if to show the message before writing it into the rmarkdown file.

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))
## leaving the default options to the code chunk
mkdCodeChunkSt(rd)
## adding a variable assignment
variable <- 1
mkdVariableAssignment(rd, "variable", "variable", show=TRUE)
mkdCodeChunkEnd(rd)

## End(Not run)
```

setBibliography	<i>setBibliography</i>
-----------------	------------------------

Description

add a bibfile name to the object that will be reflected into the report as a bibliography section

Usage

```
setBibliography(object, bibfile = NULL)
```

Arguments

object	an easyreporting class object
bibfile	a path string with the name of the bib file

Value

none

Examples

```
## Not run:
TBD

## End(Not run)
```

```
setBibliography,easyreporting-method
      setBibliography
```

Description

add a bibfile name to the object that will be reflected into the report as a bibliography section

Usage

```
## S4 method for signature 'easyreporting'
setBibliography(object, bibfile = NULL)
```

Arguments

object	an easyreporting class object
bibfile	a string with the name of the bib file

Value

none

Examples

```
## Not run:
# TBD

## End(Not run)
```

```
setOptionsList      setOptionsList
```

Description

set an optionList to the class, it automatically sets the options for the rmarkdown file. Are useful for the interpretation/execution of the code chunks.

Usage

```
setOptionsList(
  object,
  cacheFlag = TRUE,
  evalFlag = TRUE,
  echoFlag = TRUE,
  warningFlag = FALSE,
  showMessages = FALSE,
```



```

includeFlag = TRUE,
collapseFlag = FALSE,
purlFlag = TRUE,
errorFlag = TRUE,
messageFlag = TRUE,
highlightFlag = TRUE,
promptFlag = FALSE,
stripWhiteFlag = TRUE,
tidyFlag = FALSE
)

```

Arguments

object	an easyreporting class object
cacheFlag	boolean for caching chunk data (default TRUE)
evalFlag	boolean for evaluating the code chunk in the compiled version (default TRUE)
echoFlag	boolean for showing the code chunk (default TRUE)
warningFlag	boolean for showing the chunk warnings (default FALSE)
showMessages	boolean for showing the chunk warnings in compiled version (default FALSE)
includeFlag	boolean for including the code chunk in the compiled version (default TRUE)
collapseFlag	boolean for collapsing the code chunk in the compiled version (default FALSE),
purlFlag	boolean for extracting the code chunk as R code in a separate R file (default TRUE),
errorFlag	boolean for including the error generated by the code chunk in the compiled version (default TRUE),
messageFlag	boolean for including the code chunk messages in the compiled version (default TRUE),
highlightFlag	boolean for highlighting the code chunk in the compiled version (default TRUE),
promptFlag	boolean for including a ">" for the code chunk in the compiled version (default FALSE),
stripWhiteFlag	boolean for removing the white spaces at beginning/end of the code chunk in the compiled version (default TRUE),
tidyFlag	boolean for creating a tidy code chunk in the compiled version (default FALSE).

Value

none

Examples

```

## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

## setting default option

```

```
setOptionsList(rd)

## modifying only some options
rd <- setOptionsList(rd, warningFlag=TRUE,
                    showMessages=TRUE, includeFlag=TRUE)

## End(Not run)
```

setOptionsList,easyreporting-method
setOptionsList

Description

set an optionList to the class

Usage

```
## S4 method for signature 'easyreporting'
setOptionsList(
  object,
  cacheFlag = TRUE,
  evalFlag = TRUE,
  echoFlag = TRUE,
  warningFlag = FALSE,
  showMessages = FALSE,
  includeFlag = TRUE,
  collapseFlag = FALSE,
  purlFlag = TRUE,
  errorFlag = TRUE,
  messageFlag = TRUE,
  highlightFlag = TRUE,
  promptFlag = FALSE,
  stripWhiteFlag = TRUE,
  tidyFlag = FALSE
)
```

Arguments

object	an easyreporting class object
cacheFlag	boolean for caching chunk data (default TRUE)
evalFlag	boolean for evaluating the code chunk in the compiled version (default TRUE)
echoFlag	boolean for showing the code chunk (default TRUE)
warningFlag	boolean for showing the chunk warnings (default FALSE)
showMessages	boolean for showing the chunk warnings in compiled version (default FALSE)
includeFlag	boolean for including the code chunk in the compiled version (default TRUE)

collapseFlag	boolean for collapsing the code chunk in the compiled version (default FALSE),
pur1Flag	boolean for extracting the code chunk as R code in a separate R file (default TRUE),
errorFlag	boolean for including the error generated by the code chunk in the compiled version (default TRUE),
messageFlag	boolean for including the code chunk messages in the compiled version (default TRUE),
highlightFlag	boolean for highlighting the code chunk in the compiled version (default TRUE),
promptFlag	boolean for including a ">" for the code chunk in the compiled version (default FALSE),
stripWhiteFlag	boolean for removing the white spaces at beginning/end of the code chunk in the compiled version (default TRUE),
tidyFlag	boolean for creating a tidy code chunk in the compiled version (default FALSE).

Value

none

Examples

```
## Not run:
rd <- easyreporting(filenamePath="./project_report",
                    title="example_report", author=c("It's me"))

## setting default option
setOptionsList(rd)

## modifying only some options
rd <- setOptionsList(rd, warningFlag=TRUE,
                    showMessages=TRUE, includeFlag=TRUE)

## End(Not run)
```

Index

- * **internal**
 - initReportFilename, [11](#)
 - initReportFilename, easyreporting-method, [12](#)
 - mkdSetGlobalOpts, [26](#)
 - mkdSetGlobalOpts, easyreporting-method, [26](#)
 - mkdSourceFiles, [27](#)
 - mkdSourceFiles, easyreporting-method, [27](#)
 - setBibliography, easyreporting-method, [32](#)
- addResource, [3](#)
- addResource, easyreporting-method, [3](#)
- compile, [4](#)
- compile, easyreporting-method, [5](#)
- easyreporting, [5](#)
- easyreporting-class (easyreporting), [5](#)
- erGUIVolcano, [7](#)
- getBibliography, [7](#)
- getBibliography, easyreporting-method, [8](#)
- getOptionsList, [8](#)
- getOptionsList, easyreporting-method, [9](#)
- getReportFilename, [10](#)
- getReportFilename, easyreporting-method, [10](#)
- initReportFilename, [11](#)
- initReportFilename, easyreporting-method, [12](#)
- makeOptionsList, [13](#)
- mkdCodeChunkCommented, [14](#)
- mkdCodeChunkCommented, easyreporting-method, [15](#)
- mkdCodeChunkComplete, [16](#)
- mkdCodeChunkComplete, easyreporting-method, [17](#)
- mkdCodeChunkEnd, [18](#)
- mkdCodeChunkEnd, easyreporting-method, [18](#)
- mkdCodeChunkSt, [19](#)
- mkdCodeChunkSt, easyreporting-method, [20](#)
- mkdCodeChunkTitledCommented, [21](#)
- mkdCodeChunkTitledCommented, easyreporting-method, [22](#)
- mkdGeneralMsg, [23](#)
- mkdGeneralMsg, easyreporting-method, [24](#)
- mkdGeneralTitledMsg, [24](#)
- mkdGeneralTitledMsg, easyreporting-method, [25](#)
- mkdSetGlobalOpts, [26](#)
- mkdSetGlobalOpts, easyreporting-method, [26](#)
- mkdSourceFiles, [27](#)
- mkdSourceFiles, easyreporting-method, [27](#)
- mkdTitle, [28](#)
- mkdTitle, easyreporting-method, [28](#)
- mkdVariableAssignment, [29](#)
- mkdVariableAssignment, easyreporting-method, [30](#)
- person, [12](#)
- quote, [16](#), [17](#)
- setBibliography, [31](#)
- setBibliography, easyreporting-method, [32](#)
- setOptionsList, [32](#)
- setOptionsList, easyreporting-method, [34](#)