Package ‘goxygen’

August 16, 2020

Type Package
Title In-Code Documentation for ‘GAMS’
Version 1.0.3
Date 2020-08-11
Description A collection of tools which extract a model documentation from ‘GAMS’ code and comments. In order to use the package you need to install ‘pandoc’ and ‘pandoc-citeproc’ first (<https://pandoc.org/>).
Imports pander, stringi, gms, citation, yaml
URL https://github.com/pik-piam/goxygen,
https://doi.org/10.5281/zenodo.1411404
BugReports https://github.com/pik-piam/goxygen/issues
License BSD_2_clause + file LICENSE
Encoding UTF-8
LazyData no
RoxygenNote 7.1.0
Suggests testthat, knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Jan Philipp Dietrich [aut, cre],
Kristine Karstens [aut],
David Klein [aut],
Lavinia Baumstark [aut]
Maintainer Jan Philipp Dietrich <dietrich@pik-potsdam.de>
Repository CRAN
Date/Publication 2020-08-16 14:40:03 UTC
Description

helper function which adds an empty line in a markdown document

Usage

.\empty\(zz\)

Arguments

\texttt{zz} \\
\texttt{a connection object of class "textConnection" containing the markdown document}

Author(s)

Jan Philipp Dietrich

See Also

goxygen, createModulePage
Description
helper function which writes a title for a markdown section

Usage
.header(zz, title, level, id = NULL)

Arguments
zz a connection object of class "textConnection" containing the markdown document
title the title to be used (character vector of length 1)
level level of the heading (1 means main header, higher numbers reflect lower levels)
id ID given to the title (relevant for anchors)

Author(s)
Jan Philipp Dietrich

See Also
goxygen, createModulePage

Description
helper function which includes interface plot figures in a markdown document, if available. The figures need to be created beforehand.

Usage
.interfaceplot(zz, name, docfolder)

Arguments
zz a connection object of class "textConnection" containing the markdown document
name Name of the module for which the interfaceplot should be shown
docfolder folder the documentation should be written to relative to model folder
Author(s)

Jan Philipp Dietrich

See Also

goxygen, createModulePage

Description

helper function which adds a "limitations" section.

Usage

.limitations(zz, limitations, emptyIfNULL = FALSE)

Arguments

zz a connection object of class "textConnection" containing the markdown document
limitations A character vector containing the given limitations
emptyIfNULL switch which decides whether limitations section should be ignored, if limitations input is NULL or if it should state that there are no known limitations.

Author(s)

Jan Philipp Dietrich

See Also

goxygen, createModulePage
**.updateImagePaths**

Description

helper function which updates relative image paths so that they refer to a subfolder images instead of referring to the current folder.

Usage

```
.updateImagePaths(x)
```

Arguments

- `x`: A character vector containing image paths.

Author(s)

Jan Philipp Dietrich

See Also

`goxygen`, `createModulePage`

---

**.write**

Description

helper function which writes a character vector line by line in a markdown document

Usage

```
.write(zz, data)
```

Arguments

- `zz`: a connection object of class "textConnection" containing the markdown document
- `data`: a character vector to be written to the markdown document

Author(s)

Jan Philipp Dietrich

See Also

`goxygen`, `createModulePage`
buildHTML

Description

Converts a folder with markdown files and a corresponding literature library (if available) to HTML files and creates cross-links between them.

Usage

```r
buildHTML(
  style = "classic",
  folder = "html",
  mdfolder = "markdown",
  literature = "literature.bib",
  citation = "../CITATION.cff",
  supplementary = "images",
  debug = FALSE
)
```

Arguments

- **style**: visualization style to be used for the creation. Currently available styles are "classic" and "ming"
- **folder**: location the HTML files should be written to
- **mdfolder**: path to the markdown folder to be used as source
- **literature**: path to a bibliography, if available (will be ignored if file does not exist)
- **citation**: Citation information in citation file format (optional)
- **supplementary**: a vector of files and/or folders required for the conversion (e.g. an images subdirectory with figures to be shown in the documents)
- **debug**: logical which switches on/off a debug mode which will return additional status updates and keep build files

Details

Pandoc (https://pandoc.org/) together with pandoc-citeproc need to be installed on the system.

Author(s)

Jan Philipp Dietrich

See Also

goxygen, buildTEX
**buildMarkdown**

**Description**

Creates a folder filled with markdown files from a list object with markdown code.

**Usage**

```r
buildMarkdown(x, folder = "markdown")
```

**Arguments**

- `x` a named list of markdown codes which should be written as markdown files. The name of each entry will become the file name.
- `folder` folder the markdown files should be written to

**Author(s)**

Jan Philipp Dietrich

**See Also**

goxygen, buildHTML

---

**buildTEX**

**Description**

Converts a folder with markdown files and a corresponding literature library (if available) to a tex file.

**Usage**

```r
buildTEX(
  file = "documentation.tex",
  mdfolder = "markdown",
  literature = "literature.bib",
  citation = ".../CITATION.cff",
  supplementary = NULL,
  pdf = TRUE
)
```
Arguments

- **file**: name of the tex file to be written
- **mdfolder**: path to the markdown folder to be used as source
- **literature**: path to a bibliography, if available (will be ignored if file does not exist)
- **citation**: Citation information in citation file format (optional)
- **supplementary**: a vector of files and/or folders required for the conversion (e.g. an images sub-directroy with figures to be shown in the documents)
- **pdf**: boolean which specifies whether pdf file should be generated from tex

Details

Pandoc (https://pandoc.org/) together with pandoc-citeproc need to be installed on the system.

Author(s)

Jan Philipp Dietrich, Kristine Karstens

See Also

goxygen, buildHTML

Description

Support function which checks pandoc availability and stops with an error in case that pandoc cannot be found

Usage

check_pandoc(error = FALSE)

Arguments

- **error**: boolean indicating whether function should throw an error in case of missing pandoc or return a boolean FALSE.

Value

boolean indicating whether pandoc is available or not.

Author(s)

Jan Philipp Dietrich
createIndexPage

Description

Creates markdown code from a supplied data list

Usage

createIndexPage(data)

Arguments

data a list of data entries for the resulting markdown page. Following entries can be provided:

title Page title
description General description
citation A read in citation in Citation File Format (CFF)

Value

a character vector containing the produced markdown text

Author(s)

Jan Philipp Dietrich

See Also

goxygen

createListModularCode

Description

support function to create documentation of modular GAMS code.
createListSimpleCode

Usage

createListSimpleCode(
    cc,
    interfaces,
    path = ".",
    citation = NULL,
    unitPattern = c("\(\", "\)"),
    includeCore = FALSE,
    mainfile = "main.gms",
    docfolder = "doc"
)

Arguments

cc codeCheck information
interfaces interface information
path path to the model to be documented
citation citation data read from a CFF file
unitPattern pattern that is used to identify the unit in the description, default = c("\(\", ")")
includeCore Boolean whether core should be included or not, default = FALSE
mainfile main file of the model
docfolder folder the documentation should be written to relative to model folder

Author(s)

Jan Philipp Dietrich

See Also

codeCheck

description

createListSimpleCode createListSimpleCode

Description

support function to create documentation of non-modular GAMS code.

Usage

createListSimpleCode(path = ".", citation = NULL, mainfile = "main.gms")
createModulePage

Arguments

path               path to the model to be documented
citation           citation data read from a CFF file
mainfile           main file of the model

Author(s)

Jan Philipp Dietrich

See Also

codeCheck

createModulePage  createModulePage

Description

Creates markdown code from a supplied data list

Usage

createModulePage(data, docfolder)

Arguments

data               a list of data entries for the resulting markdown page. Following entries can be provided:
                     name  Name of the module
                     title Page title
                     description General description
                     input  Table containing inputs to the module
                     output Table containing outputs from the module
                     realizations A list of realizations with entries "description" and "limitations"
                     for each of them
                     declarations Table of declarations for internal objects
                     sets    Table containing sets used in the module
                     authors Module authors
                     seealso A vector with names of relevant other documentation pages.
docfolder          folder the documentation should be written to relative to model folder

Value

a character vector containing the produced markdown text
**createSimplePage**

Author(s)

Jan Philipp Dietrich

See Also

`goxygen`

Description

Creates markdown code from a supplied data list

Usage

`createSimplePage(data)`

Arguments

- **data**
  
a list of data entries for the resulting markdown page. Following entries can be provided:
  - *title*  Page title
  - *description*  General description
  - *limitations*  Limitations the implementation comes with
  - *authors*  Module authors

Value

a character vector containing the produced markdown text

Author(s)

Jan Philipp Dietrich

See Also

`goxygen`
extractDocumentation

Description

Extracts doxygen-like GAMS documentation. Entries are introduced with an @type at the beginning of the line. In case of @realization also GAMS code is read and interpreted, in all other cases only the specific documentation comment is evaluated.

Usage

extractDocumentation(path, start_type = NULL, comment = "*/")

Arguments

- **path**: path to the file(s) which should be evaluated
- **start_type**: set type for first line of code. This can be useful to extract documentation even if no documentation type has been set (e.g., reading `equations.gms` as type realization)
- **comment**: comment chars used for documentation comments

Value

- a list of documentation pieces with type as name of each element

Author(s)

Jan Philipp Dietrich

See Also

- goxygen

Examples

```r
mainfile <- paste0(system.file("dummymodel",package="gms"),"/main.gms")
calcfile <- paste0(system.file("dummymodel",package="gms"),
"/modules/02_crazymodule/complex/calculations.gms")
# extracting information from the main file of the model
extractDocumentation(mainfile)
# extracting information from a file with some equations in it
extractDocumentation(calcfile)
```
Description
Convert a gams equation into latex code

Usage
\[ \text{gamsequation2tex}(x) \]

Arguments
\( x \) GAMS equation provided as character

Value
GAMS equation converted to latex code

Author(s)
Jan Philipp Dietrich

See Also
\texttt{goxygen}

Examples
\begin{verbatim}
x <- "eq_1 .. v_a =e= \sum_{j}(v_b(j)*(1-s_c)+\sum_{cell(i,j)}(v_d(i)/f_d(i))));"
cat(gamsequation2tex(x))
\end{verbatim}

Description
Documentation function which extracts a full model documentation from a modularized gams model. The function extracts comments used as documentation, extracts code and can extract and convert GAMS equations as latex code. Output is returned in Markdown, HTML and PDF format.
goxygen

Usage

```r
library(goxygen)

# Example usage
# goxygen(
#   path = ".",
#   docfolder = "doc",
#   cache = FALSE,
#   output = c("html", "pdf"),
#   htmlStyle = "ming",
#   cff = "CITATION.cff",
#   modularCode = is.modularGAMS(),
#   unitPattern = c("\((\", "\))"),
#   includeCore = FALSE,
#   mainfile = "main.gms",
#   ...
# )
```

Arguments

- `path`: path to the model to be documented
- `docfolder`: folder the documentation should be written to relative to model folder
- `cache`: Boolean to allow read data from existing cache file
- `output`: List of output to be written, available are "html", "pdf" and "tex"
- `htmlStyle`: visualization style to be used for the HTML creation. Currently available styles are "classic" and "ming". Ignored for outputs other than HTML.
- `cff`: path to a citation file in citation-file-format (ignored if not existing)
- `modularCode`: Boolean deciding whether code should be interpreted as modular GAMS code (only av)
- `unitPattern`: pattern that is used to identify the unit in the description, default =c("\(", ")")
- `includeCore`: boolean whether core should be included or not, default=FALSE
- `mainfile`: main file of the model
- `...`: optional arguments to `interfaceplot`, passed via `modules_interfaceplot`.

Note

Documentation lines in the code must start with 
 to be detected as documentation. Identifier at the beginning of each block describe what kind of documentation is given. All identifiers start with @ followed by the name of the identifier. Currently, following identifiers are available

- `@title`: Title
- `@authors`: List of authors
- `@description`: Model description (only the documentation text will be interpreted)
- `@equations`: Equation description (documentation text will be extracted and gams equations will be converted to latex code)
- `@code`: Code description (documentation text and code will be extracted)
- `@limitations`: details about limitations of an implementation
• @stop everything following will be ignored until the next identifier is mentioned again. Useful to stop a section

In addition you can store a model logo (100px height, 100px weight) as logo.png in the main folder of the model which then will be used in the HTML version of the documentation. If you want to add citations to your documentation you can do so by adding a bibtex file with the name literature.bib in the main folder of the model. To link these references in the text you can use the syntax @<id> in which "<id>" stands for the identifier given to the corresponding bibtex entry.

Author(s)

Jan Philipp Dietrich

See Also
codeCheck, interfaceplot

Examples

```r
# make sure that pandoc is available
if(check_pandoc()) {
  # run goxygen for dummy model and store documentation as HTML in a temporary directory
docfolder <- paste0(tempdir(),"/doc")
goxygen(system.file("dummymodel",package="gms"), docfolder=docfolder, output="html")
}
```

Description

Converts a folder with markdown files and a corresponding literature library (if available) to HTML files and creates cross-links between them.

Usage

```r
oldBuildHTML(
  folder = "html",
  mdfolder = "markdown",
  literature = "literature.bib",
  citation = "..://CITATION.cff",
  supplementary = NULL,
  addHTML = NULL
)
```
returnReferences

Arguments

folder location the HTML files should be written to
mdfolder path to the markdown folder to be used as source
literature path to a bibliography, if available (will be ignored if file does not exist)
citation Citation information in citation file format (optional)
supplementary a vector of files and/or folders required for the conversion (e.g. an images subdirectory with figures to be shown in the documents)
addHTML character vector with HTML code which should be added to the body of each HTML file.

Details

Pandoc (https://pandoc.org/) together with pandoc-citeproc need to be installed on the system.

Author(s)

Jan Philipp Dietrich

See Also

goxygen, buildTEX

Description

Support function to create a reference file linking references with corresponding addresses.

Usage

returnReferences(names, targets, file, level = 2)

Arguments

names vector of reference names
targets vector of reference addresses (same order and lengths as names)
file name of the reference file to be written
level level of the "References" title to be written

Author(s)

Jan Philipp Dietrich
Index

.empty, 2
.header, 3
.interfaceplot, 3
.limitations, 4
.updateImagePaths, 5
.write, 5

buildHTML, 6, 7, 8
buildMarkdown, 7
buildTEX, 6, 7, 17

docCheck, 10, 11, 16
createIndexPage, 9
createListModularCode, 9
createListSimpleCode, 10
createModulePage, 2–5, 11
createSimplePage, 12

eXtractDocumentation, 13

gamsequation2tex, 14
goxygen, 2–9, 12–14, 14, 17

.interfaceplot, 15, 16

modules_interfaceplot, 15

oldBuildHTML, 16

returnReferences, 17