

# Package ‘wyz.code.rdoc’

October 8, 2019

**Type** Package

**Title** Wizardry Code Offensive Programming R Documentation

**Version** 1.1.7

**Author** Fabien Gelineau <neonira@gmail.com>

**Maintainer** Fabien Gelineau <neonira@gmail.com>

**Description** Allows to generate automatically R documentation files from offensive programming test cases. It populates most of the section of the documentation content from the offensive programming instrumentation. This reduces greatly the package producer effort and time to get to a fully documented manual page for any instrumented function. Following documentation sections are now automatically filled from instrumentation data: title, description, usage, arguments, value, author, examples. Sections references, notes and keyword are instrumented to industrialize their production. Produced manual pages are ready for completion (e.g note section if needed), language and phrasal adjustments. Main task for the package producer is now review, no more content production. Refer to chapter 11 of Offensive Programming Book, Fabien GELINEAU (2019, ISBN:979-10-699-4075-8), to learn about details and get value from this package.

**Encoding** UTF-8

**LazyData** true

**License** GPL-3

**Depends** R (>= 3.5)

**Imports** methods, data.table (>= 1.11.8), tidyr, lubridate (>= 1.7.4),  
wyz.code.offensiveProgramming (>= 1.1.12), R6 (>= 2.4.0)

**Suggests** testthat, knitr, rmarkdown

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-10-08 08:10:02 UTC

**R topics documented:**

generateContent . . . . .	2
generateDocumentationContent . . . . .	3
generateDocumentationFile . . . . .	4
generateReference . . . . .	5
generateS3MethodSignature . . . . .	6
generateSection . . . . .	7
generateSpecialLink . . . . .	8
generateWrapperObject . . . . .	9
getObjectSignature . . . . .	10
rdocKeywords . . . . .	10
retrievePackageFunctionNames . . . . .	11
sentensize . . . . .	12
WrapperObject . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

generateContent	<i>Generate R documentation content</i>
-----------------	---

---

**Description**

Generate a R documentation atomic piece

**Usage**

```
generateContent(content_s, keyword_s_1 = NA, content2_s = NA,
               inline_b_1 = TRUE, useSpace_b_1 = FALSE)
```

**Arguments**

content_s	the content to consider
keyword_s_1	a R documentation keyword. See <a href="#">rdocKeywords</a> .
content2_s	a second content, useful with some keywords that require two members
inline_b_1	should the printed result be inline or not ?
useSpace_b_1	when dealing with documentation keywords that requires two members, some may require a space in between to work properly. This parameters allows you to ask for this.

**Details**

Very convenient function, to customize your R documentation output.

Might be used programmatically to generate pieces or full documentation.

See examples below.

**Value**

A string, containing one or several lines of text.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(generateContent('a title', 'title'))
print(generateContent('https://neonira.github.io/offensiveProgrammingBook/', 'href',
  'Offensive Programming Book'))
print(generateContent('a', 'item', 'description of a', useSpace_b_1 = TRUE))
print(generateContent('a', 'item', 'description of a', useSpace_b_1 = FALSE))
```

---

generateDocumentationContent

*Generate R documentation file.*

---

**Description**

Generate a R documentation file from an offensive programming instrumented object.

**Usage**

```
generateDocumentationContent(targetFolder_s_1, kind_s_1, name_s_1,
  object_o_1, packageName_s_1, extraneous_l = list(),
  typeFactory_o_1 = FunctionParameterTypeFactory(),
  overwrite_b_1 = FALSE)
```

**Arguments**

targetFolder_s_1	the target folder to write files to
kind_s_1	a single string taken amongst 'package', 'class', 'method'.
name_s_1	Either the package name, the class name or the method name, depending on previous argument.
object_o_1	an object of the given class
packageName_s_1	the name of the targeted package to produce documentation for
extraneous_l	a list of extraneous data, to consider when producing R documentation content.

typeFactory\_o\_1 a type factory object. See [FunctionParameterTypeFactory](#)  
overwrite\_b\_1 a boolean, allowing file overwriting when file already exists.

### Value

A list with two entries named filename and overwritten, expressing the filename and the overwritten boolean flag.

If a single file is produced, then a such list is returned, otherwise, a list of a such list is returned.

### Note

Allows to produce on-demand, 3 kinds of R documentation files, according to passed parameters.

To produce, all the methods files, in a single shot, from the provided R object, simply pass NA as parameter name\_s\_1 and 'method' as parameter kind\_s\_1.

### Author(s)

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

### References

Refer to [Writing R extensions](#) to know more about R documentation requirements.

### Examples

```
# see system.file('inst/ut-generateDocumentationFiles.R', package = 'wyz.code.rdoc')  
# for a complete example
```

---

generateDocumentationFile  
*Generate R documentation file*

---

### Description

Write documentation file on disk.

### Usage

```
generateDocumentationFile(filename_s_1, content_s,  
                          overwrite_b_1 = FALSE, verbose_b_1 = TRUE)
```

**Arguments**

filename\_s\_1 the file name to write. Might be a relative or absolute file path. Suffix .Rd will be added if missing.

content\_s the content to write into the file, given as a vector of strings.

overwrite\_b\_1 a boolean, allowing file overwriting when file already exists.

verbose\_b\_1 a boolean to show a message when file is overwritten.

**Value**

A list with two entries named filename and overwritten, expressing the file name and the overwritten boolean flag.

**Note**

Should be used when programming your own way to produce R documentation.

As a end-user, no need to use this function. Use [generateDocumentationContent](#) instead.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
content <- c(
  generateSection('name', 'alpha'),
  generateSection('alias', 'alpha'),
  generateSection('title', 'alpha function'),
  generateSection('description', 'Function alpha allows to ...')
)
generateDocumentationFile(file.path(tempdir(), 'myRdoc'), content, TRUE)
```

---

generateReference      *Generate R documentation reference piece*

---

**Description**

Allows to generate content to be used in references R documentation part (or elsewhere), from a list.

**Usage**

```
generateReference(data_1)
```

**Arguments**

data\_l            a list with names url, label, comment

**Details**

Generates a R documentation \href entry. The url holds the URL part, the label holds the label part, and the comment if provided, is appended to the result. The resulting string is appended to the string 'Refer to'.

**Value**

A single string.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(generateReference(list(url = 'https://neonira.github.io/offensiveProgrammingBook/',
                             label = 'Offensive Programming Book')))
# provides following result
# "Refer to \href{https://neonira.github.io/offensiveProgrammingBook/}{Offensive Programming Book}."
```

---

generateS3MethodSignature

*Generate S3 method signature*

---

**Description**

Function to create easily function signature from an S3 class

**Usage**

```
generateS3MethodSignature(methodName_s_1, className_s_1, argumentNames_s)
```

**Arguments**

methodName\_s\_1    the function/method name to consider  
 className\_s\_1    the class name to consider  
 argumentNames\_s    the function/method argument names

**Value**

A single string

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(generateReference(list(url = 'https://neonira.github.io/offensiveProgrammingBook/',
                             label = 'Offensive Programming Book')))
# provides following result
# "Refer to \href{https://neonira.github.io/offensiveProgrammingBook/}{Offensive Programming Book}."
```

---

generateSection	<i>Generate a R documentation section</i>
-----------------	---

---

**Description**

Generate a R documentation section

**Usage**

```
generateSection(sectionName_s_1, content_s)
```

**Arguments**

sectionName_s_1	the section name to consider. Must be known from <a href="#">rdocKeywords</a> .
content_s	the section textual content to consider

**Details**

Sections inlining is automated and managed internally. Do not bother about it.

**Value**

A single string.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(generateReference(list(url = 'https://neonira.github.io/offensiveProgrammingBook/',
                             label = 'Offensive Programming Book')))
# provides following result
# "Refer to \href{https://neonira.github.io/offensiveProgrammingBook/}{Offensive Programming Book}."
```

---

generateSpecialLink    *Generate R documentation special link*

---

**Description**

Will create a special R documentation link on package name, using rare syntax `\link[packagename:topicname]{packagename:topicname}`.

**Usage**

```
generateSpecialLink(packageName_s_1, topicName_s_1)
```

**Arguments**

```
packageName_s_1    the package name to consider
topicName_s_1    the topic name to consider
```

**Value**

A single string.

**Note**

It is not possible to create such R documentation special link, using [generateContent](#) function.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>  
 Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(generateSpecialLink('wyz.doc.offensiveProgramming', 'runTestCase'))
```



---

generateWrapperObject *Generate wrapper object.*

---

## Description

Use method generateWrapperObject to generate a wrapper object.

## Usage

```
generateWrapperObject(fun_f_1, functionName_ch_1 = NA)
```

## Arguments

fun\_f\_1            A 1-length vector holding a function, to be encapsulated into the object.  
functionName\_ch\_1    A 1-length vector of character values, expressing the function name.

## Value

An object of class `WrapperObject` that could be used to easily generate manual pages using `wyz.code.rdoc`.

## Author(s)

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

## Examples

```
# typical case
myfun <- function(x_i, y_i) { x_i + y_i }
o <- generateWrapperObject(myfun)
print(o)
# <environment: 0x0000021d7abd9710>
# attr(,"class")
# [1] "WrapperObject" "environment"
ls(o)
# [1] "removeEllipsisName" "self"
```

getObjectSignature      *Get Object Signature*

---

**Description**

Get object reification code as string

**Usage**

```
getObjectSignature(object_o_1)
```

**Arguments**

object\_o\_1      the R object to analyze

**Value**

A single string.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**Examples**

```
source(file.path(system.file(package = "wyz.code.offensiveProgramming"), 'code-samples',
'classes', 'sample-classes.R'))
getObjectSignature(Wyx())
#[1] "Wyx(d_d, y_b_1 = FALSE)"
```

---

rdocKeywords      *R documentation keywords*

---

**Description**

Provide R documentation keywords inventory

**Usage**

```
rdocKeywords(asList_b_1 = FALSE)
```

**Arguments**

asList\_b\_1      a boolean to specify desired output

**Value**

A vector when parameter `asList_b_1` is FALSE, a list otherwise.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**References**

Refer to [Writing R extensions](#) to know more about R documentation requirements.

**Examples**

```
print(rdocKeywords())
print(rdocKeywords(TRUE))
```

---

retrievePackageFunctionNames

*Retrieve Package Function Names*

---

**Description**

Get the function names from a package name

**Usage**

```
retrievePackageFunctionNames(packageName_s_1, libraryPath_s_1 = .libPaths()[1])
```

**Arguments**

packageName\_s\_1

a single string, that is the package name to seek for

libraryPath\_s\_1

the folder to look for

**Value**

A vector of function names

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**Examples**

```
retrievePackageFunctionNames('wyz.code.rdoc')
```

---

`sentensize`*Sentensize a string*

---

### Description

Transforms a string into sentence, enforcing upper case on first letter, normalizing spaces, and ensuring a final point is present in the sentence. All spaces characters are turned to single space.

### Usage

```
sentensize(x_s_1, ...)
```

### Arguments

<code>x_s_1</code>	the string to sentensize.
<code>...</code>	any complimentary string. Will be appended to <code>x_s_1</code> prior processing

### Details

Intended to produce well-formed sentences, human readable (from a format point of view).

### Value

A single string.

### Author(s)

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

### References

Refer to [Writing R extensions](#) to know more about R documentation requirements.

### Examples

```
print(sentensize('some bla', 'bli blo \t\n\bblu'))

# provides following result
# "Some bla bli blo blu."
```

---

WrapperObject	<i>Wrapper object.</i>
---------------	------------------------

---

**Description**

Class WrapperObject from package `wyz.code.metaTesting`.

**Usage**

```
WrapperObject()
```

**Value**

An object of class WrapperObject.

**Note**

This object is intended for programmers, not for end-users.

**Author(s)**

Fabien Gelineau <neonira@gmail.com>

Maintainer: Fabien Gelineau <neonira@gmail.com>

**Examples**

```
# typical use case  
wo <- WrapperObject()
```

# Index

## \*Topic **documentation**

- [generateContent](#), 2
- [generateDocumentationContent](#), 3
- [generateDocumentationFile](#), 4
- [generateReference](#), 5
- [generateS3MethodSignature](#), 6
- [generateSection](#), 7
- [generateSpecialLink](#), 8
- [getObjectSignature](#), 10
- [rdocKeywords](#), 10
- [sentensize](#), 12

## \*Topic **keywords**

- [generateContent](#), 2
- [generateDocumentationContent](#), 3
- [generateDocumentationFile](#), 4
- [generateReference](#), 5
- [generateS3MethodSignature](#), 6
- [generateSection](#), 7
- [generateSpecialLink](#), 8
- [getObjectSignature](#), 10
- [rdocKeywords](#), 10
- [sentensize](#), 12

[FunctionParameterTypeFactory](#), 4

- [generateContent](#), 2, 8
- [generateDocumentationContent](#), 3, 5
- [generateDocumentationFile](#), 4
- [generateReference](#), 5
- [generateS3MethodSignature](#), 6
- [generateSection](#), 7
- [generateSpecialLink](#), 8
- [generateWrapperObject](#), 9
- [getObjectSignature](#), 10

[rdocKeywords](#), 2, 7, 10

[retrievePackageFunctionNames](#), 11

[sentensize](#), 12

[WrapperObject](#), 9, 13