

Package: docket (via r-universe)

September 11, 2024

Title Insert R Data into 'Word' Documents

Version 1.20

Description Populate data from an R environment into '.doc' and '.docx' templates. Create a template document in a program such as 'Word', and add strings encased in guillemet characters to create flags («example»). Use `getDictionary()` to create a dictionary of flags and replacement values, then call `docket()` to generate a populated document.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports stringr, XML, xml2, zip

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/JonathanConrad98/docket>

BugReports <https://github.com/JonathanConrad98/docket/issues>

Config/testthat/edition 3

Repository <https://jonathanconrad98.r-universe.dev>

RemoteUrl <https://github.com/jonathanconrad98/docket>

RemoteRef HEAD

RemoteSha 9e5adad29e824f70e900930b42c46a1c1314a4a2

Contents

batchDocket	2
checkBatchDictionary	3
checkDictionary	4
docket	5
getBatchDictionary	6
getDictionary	7

batchDocket *Create Documents*

Description

Scans the input template file for specified flags as defined in the dictionary, and replaces them with corresponding data. Repeats the process for each column, generating a new document for each column which is saved as the file name and path listed in row 1

Usage

```
batchDocket(filename, batchDictionary)
```

Arguments

`filename` The file path to the document template. Supports .doc and .docx formats

`batchDictionary` A data frame where each row represents a flag to be replaced in the template document and each column represents a final document to be generated

Value

Generates new .doc or .docx files with the flags replaced by the specified data for that column

Examples

```
# Path to the sample template file included in the package
template_path <- system.file("batch_document", "batchTemplate.docx", package="docket")
output_paths <- as.list(paste0(dirname(template_path), paste0("/batch document", 1:5, ".docx")))

# Create a dictionary by using the getDictionary function on the sample template file
result <- getBatchDictionary(template_path, output_paths)
result[2,2:ncol(result)] <- Sys.getenv("USERNAME") #Author name
result[3,2:ncol(result)] <- as.character(Sys.Date())
result[4,2:ncol(result)] <- 123
result[5,2:ncol(result)] <- 456
result[6,2:ncol(result)] <- 789
result[7,2:ncol(result)] <- sum(as.numeric(result[4:6,2]))

# Verify that the result dictionary is valid
if (checkBatchDictionary(result) == TRUE) {
  batchDocket(template_path, result)
  for (i in 1:length(output_paths)) {
    if (file.exists(output_paths[[i]])) {
      print(paste("docket", i, "Successfully Created"))
    }
  }
}
```

checkBatchDictionary *Check that the batch dictionary is valid*

Description

Validates that the input batch dictionary meets the following requirements: #' #'

- **1.** It is a data frame
- **2.** Column 1 is named "flag"
- **3.** Column 1 contains flags with starting and ending wings: « »
- **4.** Row 1 contains the file names and paths of the populated output documents

Usage

```
checkBatchDictionary(batchDictionary)
```

Arguments

batchDictionary

A data frame where each row represents a flag to be replaced in the template document and each column represents a final document to be generated

Value

Logical. Returns 'TRUE' if the batch dictionary meets requirements for processing. Returns 'FALSE' otherwise

Examples

```
# Path to the sample template file included in the package
template_path <- system.file("batch_document", "batchTemplate.docx", package="docket")
output_paths <- as.list(paste0(dirname(template_path), paste0("/batch document", 1:5, ".docx")))

# Create a dictionary by using the getDictionary function on the sample template file
result <- getBatchDictionary(template_path, output_paths)
result[2,2:ncol(result)] <- Sys.getenv("USERNAME") #Author name
result[3,2:ncol(result)] <- as.character(Sys.Date())
result[4,2:ncol(result)] <- 123
result[5,2:ncol(result)] <- 456
result[6,2:ncol(result)] <- 789
result[7,2:ncol(result)] <- sum(as.numeric(result[4:6,2]))

# Verify that the result dictionary is valid
if (checkBatchDictionary(result) == TRUE) {
  print("Valid Batch Dictionary")
}
```

checkDictionary *Check if dictionary meets specific requirements.*

Description

Verifies that the input dictionary meets the following conditions #'

- **1.** It is a two-column data frame
- **2.** Column 1 is named "flag"
- **3.** Column 1 contains flags with starting and ending wings: « »

Usage

```
checkDictionary(dictionary)
```

Arguments

dictionary A data frame where each row represents a flag in the template document and its replacement value

Value

Logical. Returns 'TRUE' if the dictionary meets requirements for processing. Returns 'FALSE' otherwise

Examples

```
# Path to the sample template included in the package
template_path <- system.file("template_document", "Template.docx", package="docket")

# Create a dictionary by using the getDictionary function on the sample template file
result <- getDictionary(template_path)

# Insert data into the template dictionary
result[1,2] <- Sys.getenv("USERNAME") #Author name
result[2,2] <- as.character(Sys.Date()) # Date report created
result[3,2] <- 123
result[4,2] <- 456
result[5,2] <- 789
result[6,2] <- sum(as.numeric(result[3:5,2]))

# Verify that the result dictionary is valid
if (checkDictionary(result) == TRUE) {
  print("Valid Dictionary")
}
```

`docket`*Create documents*

Description

Scans the input template file for specified flags as defined in the dictionary, and replaces them with corresponding data. The edited content is then saved to a new document

Usage

```
docket(filename, dictionary, outputName)
```

Arguments

<code>filename</code>	The file path to the document template
<code>dictionary</code>	A data frame where each row represents a flag in the template document and its replacement value
<code>outputName</code>	The file path and name for the saved output document

Value

Generates a new .doc or .docx file with the flags replaced by the specified data

Examples

```
# Path to the sample template included in the package
template_path <- system.file("template_document", "Template.docx", package="docket")
output_path <- paste0(dirname(template_path), "/output document.docx")

# Create a dictionary by using the getDictionary function on the sample template file
result <- getDictionary(template_path)

# Insert data into the template dictionary
result[1,2] <- Sys.getenv("USERNAME") #Author name
result[2,2] <- as.character(Sys.Date()) # Date report created
result[3,2] <- 123
result[4,2] <- 456
result[5,2] <- 789
result[6,2] <- sum(as.numeric(result[3:5,2]))

# Verify that the result dictionary is valid
if (checkDictionary(result) == TRUE) {
  docket(template_path, result, output_path)
  if (file.exists(output_path)) {
    print("Docket Successfully Created")
  }
}
```

getBatchDictionary *Dictionary for multiple docket outputs*

Description

Scans the input file for strings enclosed by flag wings: « ». Creates a replacement value column for each document to be generated

Usage

```
getBatchDictionary(  
  filename,  
  outputFiles,  
  dictionaryLength = length(outputFiles)  
)
```

Arguments

filename The file path to the document template

outputFiles A list of the file names and paths for the populated templates

dictionaryLength Number of columns in the batch dictionary. Defaults to the number of output files. Cannot be shorter than the count of 'outputFiles'

Value

Data frame for populating data into the template with row 1 containing the output file names

Examples

```
# Path to the sample template file included in the package  
template_path <- system.file("batch_document", "batchTemplate.docx", package="docket")  
output_paths <- as.list(paste0(dirname(template_path), paste0("/batch document", 1:5, ".docx"))) )  
  
# Create a dictionary by using the getDictionary function on the sample template file  
result <- getBatchDictionary(template_path, output_paths)  
print(result)
```

getDictionary	<i>Create a dictionary</i>
---------------	----------------------------

Description

Scans the input file for strings enclosed by flag wings: « », then creates an empty dictionary with corresponding replacement values for each flag.

Usage

```
getDictionary(filename)
```

Arguments

filename The file path to the document template. Supports .doc and .docx

Value

A data frame where each row represents a flag in the template document and its replacement value

Examples

```
# Path to the sample template file included in the package
template_path <- system.file("template_document", "Template.docx", package="docket")

# Create a dictionary by using the getDictionary function on the sample template file
result <- getDictionary(template_path)
print(result)
```

Index

`batchDocket`, [2](#)

`checkBatchDictionary`, [3](#)

`checkDictionary`, [4](#)

`docket`, [5](#)

`getBatchDictionary`, [6](#)

`getDictionary`, [7](#)