

Package: kwb.endnote (via r-universe)

August 21, 2024

Title Helper Functions for Analysing KWB Endnote Library (Exported as .xml)

Version 0.2.0

Description Helper Functions For Analysing KWB Endnote Library (Exported As .XML).

License MIT + file LICENSE

URL <https://github.com/KWB-R/kwb.endnote>

BugReports <https://github.com/KWB-R/kwb.endnote/issues>

Imports dplyr, forcats, fs, ggplot2, kwb.file, kwb.read, kwb.utils, magrittr, openxlsx, rlang, stringr, tibble, tidyr, wordcloud2, xml2, plotly

Suggests covr, knitr, rmarkdown

Remotes github::kwb-r/kwb.file, github::kwb-r/kwb.read, github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

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

RemoteUrl <https://github.com/KWB-R/kwb.endnote>

RemoteRef HEAD

RemoteSha a170fb4f501f97ed056cb62e67cc4716721286c6

Contents

add_file_info_attributes	2
check_for_differences	3
check_problematic_entries	4
clean_accessibility	5
clean_author_names	5

clean_dois	6
clean_project_names	6
clean_references_df	7
create_df_from_endnote_xml	7
create_endnote_list	8
create_keywords_df	8
create_list_by_pubtype_from_df	9
create_list_with_unique_entries	10
create_references_df	10
default_clean_xlsx	11
default_xlsx	11
default_xml	12
extdata_file	12
get_abstract	12
get_authors	13
get_available_multi_cols	13
get_keywords	14
get_pdfurls	14
get_reference_type_names	15
get_secondary_authors	15
get_tertiary_authors	16
get_xml_filename_without_extension	16
give_hints_accessibility	17
give_hints_author_names	17
give_hints_project_names	18
plot_pubs_by_author	18
plot_pubs_by_year	19
plot_wordcloud_keywords	19
record_list_to_df	20
tidy_df	20
tidy_multi_cols_df	21
tidy_multi_cols_list	22
tidy_selected_cols	22
write_clean_references_df_to_xlsx	23
write_references_df_to_xlsx	23

Index **25**

add_file_info_attributes

Helper function: add fileinfo attributes

Description

Helper function: add fileinfo attributes

Usage

```
add_file_info_attributes(obj, path)
```

Arguments

```
obj          object to write attributes to
path        path to file
```

Value

object with file info attributes

check_for_differences *Check two Dataframes for Differences*

Description

Check two Dataframes for Differences

Usage

```
check_for_differences(df_x, df_y, dbg = TRUE)
```

Arguments

```
df_x        data frame as retrieved by create_references_df() or clean_references_df()
df_y        data frame as retrieved by create_references_df() or clean_references_df()
dbg         should dbg messages be printed (default: TRUE)
```

Value

a dataframe containing only the differences between df_x and df_y

Examples

```
## Not run:
#####
### Option 1
### Check differences between two different versions of "KWB_documents.xml"
#####

old_xml <- extdata_file("2020-05-25_KWB-documents.xml")
new_xml <- extdata_file("2020-06-17_KWB-documents.xml")
old_list <- kwb.endnote::create_endnote_list(old_xml)
new_list <- kwb.endnote::create_endnote_list(new_xml)
old_df <- kwb.endnote::create_references_df(old_list)
new_df <- kwb.endnote::create_references_df(new_list)
diffs_df_oldnew <- check_for_differences(old_df, new_df)
```

```

head(diffs_df_oldnew)

#####
### Option 2:
### Check differences between "as-is" import and "collapsing" fields
#####

endnote_list <- create_endnote_list()
refs_df <- create_references_df(endnote_list)
refs_df_collapse <- create_references_df(endnote_list, collapse = TRUE)
diffs_df <- check_for_differences(refs_df, refs_df_collapse)
head(diffs_df)

## End(Not run)

```

```

check_problematic_entries
      Check Endnote for Problematic References

```

Description

Check Endnote for Problematic References

Usage

```
check_problematic_entries(endnote_list, give_hints = TRUE, dbg = TRUE)
```

Arguments

endnote_list	list created with create_endnote_list()
give_hints	if TRUE hints will be generated, e.g. "add_public_or_confidential" for accessibility data
dbg	show debug messages (default: TRUE)

Value

a data frame with problematic entries

Examples

```

## Not run: endnote_list <- create_endnote_list()
problematic_entries <- check_problematic_entries(endnote_list)
head(problematic_entries)

## End(Not run)

```

clean_accessibility *Helper function: clean access information*

Description

Helper function: clean access information

Usage

```
clean_accessibility(access, give_hints = FALSE, dbg = TRUE)
```

Arguments

access	vector with accessibility information
give_hints	if TRUE hints will be generated, e.g. "add_public_or_confidential" in case of missing entries (default: FALSE)
dbg	show debug messages (default: TRUE)

Value

vector with cleaned accessibility information

clean_author_names *Helper function: clean author names*

Description

Helper function: clean author names

Usage

```
clean_author_names(author_names, give_hints = FALSE, dbg = TRUE)
```

Arguments

author_names	with author names to clean
give_hints	if TRUE hints will be generated, e.g. "fix_multiple_authors_per_line" in case of missing entries (default: FALSE)
dbg	show debug messages (default: TRUE)

Value

vector with cleaned author names

clean_dois *Helper function: clean DOIs*

Description

Helper function: clean DOIs

Usage

```
clean_dois(dois, dbg = TRUE)
```

Arguments

dois	vectors with DOIs to clean
dbg	show debug messages (default: TRUE)

Value

cleaned DOIs

clean_project_names *Helper function: clean project names*

Description

Helper function: clean project names

Usage

```
clean_project_names(project_names, give_hints = FALSE, dbg = TRUE)
```

Arguments

project_names	with project names to clean
give_hints	if TRUE hints will be generated, e.g. "add_project_name" in case of missing entries (default: FALSE)
dbg	show debug messages (default: TRUE)

Value

vector with cleaned project names

clean_references_df *Clean References Dataframe*

Description

Clean References Dataframe

Usage

```
clean_references_df(endnote_list, give_hints = FALSE, dbg = TRUE)
```

Arguments

endnote_list	list created with create_endnote_list()
give_hints	if TRUE hints will be generated, e.g. "add_public_or_confidential" for accessibility data
dbg	show debug messages (default: TRUE)

Value

cleaned references_df

Examples

```
## Not run: endnote_list <- create_endnote_list()
refs_clean_df <- clean_references_df(endnote_list)
head(refs_clean_df)

## End(Not run)
```

create_df_from_endnote_xml
 Create dataframe from Endnote XML file

Description

Create dataframe from Endnote XML file

Usage

```
create_df_from_endnote_xml(endnote_xml = default_xml())
```

Arguments

endnote_xml	path to Endnote library exported as .xml (default: default_xml())
-------------	---

Value

data.frame with all information from Endnote XML joined with data from get_reference_type_names()

Examples

```
references_df <- create_df_from_endnote_xml()
head(references_df)
```

create_endnote_list *Create List From Endnote XML*

Description

Create List From Endnote XML

Usage

```
create_endnote_list(endnote_xml = default_xml())
```

Arguments

endnote_xml path to Endnote library exported as .xml (default: default_xml())

Value

list of imported endnote_xml

Examples

```
endnote_list <- create_endnote_list()
str(endnote_list[1]$record)
attr(endnote_list, "xml_file_info")
attr(endnote_list, "xml_filename_without_extension")
```

create_keywords_df *Create Keywords Dataframe*

Description

Create Keywords Dataframe

Usage

```
create_keywords_df(references_df)
```


Arguments

`references_df` `references_df` as recieved from `kwb.endnote::create_df_from_endnote_xml()`

Value

keywords dataframe

Examples

```
references_df <- create_df_from_endnote_xml()
keywords_df <- create_keywords_df(references_df)
head(keywords_df)
```

```
create_list_by_pubtype_from_df
      Create List By Pub Type From Datafram
```

Description

Create List By Pub Type From Datafram

Usage

```
create_list_by_pubtype_from_df(refs_df)
```

Arguments

`refs_df` data frame as created with `create_references_df()`

Value

list with references with one sublist for each publication type

Examples

```
## Not run:
endnote_list <- create_endnote_list()
refs_df <- create_references_df(endnote_list)
refs_list_by_pubtype <- create_list_by_pubtype_from_df(refs_df)
str(refs_list_by_pubtype, 1)

## End(Not run)
```

```
create_list_with_unique_entries
```

Create List with Unique Entries

Description

Create List with Unique Entries

Usage

```
create_list_with_unique_entries(refs_df)
```

Arguments

refs_df data frame as created with create_references_df()

Value

list with unique values for selected columns

Examples

```
## Not run: endnote_list <- create_endnote_list()
refs_df <- create_references_df(endnote_list)
unique_entries_list <- create_list_with_unique_entries(refs_df)
str(unique_entries_list, 1)

## End(Not run)
```

```
create_references_df    Create References Dataframe
```

Description

Create References Dataframe

Usage

```
create_references_df(endnote_list, collapse = FALSE)
```

Arguments

endnote_list list created with create_endnote_list()
collapse should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

data.frame with columns record_id, rec_number, ref_type_id, ref_type_name

Examples

```
endnote_list <- create_endnote_list()
refs_df <- create_references_df(endnote_list)
head(refs_df)
```

default_clean_xlsx *Helper function: default filename for cleaned XLSX*

Description

Helper function: default filename for cleaned XLSX

Usage

```
default_clean_xlsx(endnote_list)
```

Arguments

endnote_list list as retrieved by create_endnote_list()

Value

default clean xlsx filename

default_xlsx *Helper function: default filename for XLSX*

Description

Helper function: default filename for XLSX

Usage

```
default_xlsx(endnote_list)
```

Arguments

endnote_list list as retrieved by create_endnote_list()

Value

default xlsx filename

default_xml	<i>Path to Default XML File</i>
-------------	---------------------------------

Description

Path to Default XML File

Usage

```
default_xml()
```

Value

path to xml file stored in this package, containing references from KWB Endnote database

extdata_file	<i>Get Path to File in This Package</i>
--------------	---

Description

Get Path to File in This Package

Usage

```
extdata_file(...)
```

Arguments

... parts of path passed to [system.file](#)

get_abstract	<i>Helper function: get abstract from list for a reference</i>
--------------	--

Description

Helper function: get abstract from list for a reference

Usage

```
get_abstract(record_list, collapse = FALSE)
```

Arguments

record_list	list with one record of <code>create_endnote_list()</code>
collapse	should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row abstract data frame

get_authors	<i>Helper function: get authors from list for a reference</i>
-------------	---

Description

Helper function: get authors from list for a reference

Usage

```
get_authors(
  record_list,
  col_name = "author",
  extract_value = "authors",
  collapse = FALSE
)
```

Arguments

record_list	list with one record of create_endnote_list()
col_name	default: "author"
extract_value	extract_value = "authors"
collapse	should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row authors data frame

get_available_multi_cols	<i>Helper Function: Get Available Multi Col Names</i>
--------------------------	---

Description

all names which are valid inputs for tidy_selected_cols()

Usage

```
get_available_multi_cols(df)
```

Arguments

df	as retrieved by create_references_df() or clean_references_df()
----	---

Value

all names which are valid inputs for tidy_selected_cols()

get_keywords	<i>Helper function: get keywords from list for a reference</i>
--------------	--

Description

Helper function: get keywords from list for a reference

Usage

```
get_keywords(
  record_list,
  col_name = "keyword",
  extract_value = "keywords",
  collapse = FALSE
)
```

Arguments

record_list	list with one record of create_endnote_list()
col_name	default: "keyword"
extract_value	extract_value = "keyword"
collapse	should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row keywords data frame

get_pdfurls	<i>Helper function: get pdfurls from list for a reference</i>
-------------	---

Description

Helper function: get pdfurls from list for a reference

Usage

```
get_pdfurls(record_list, col_name = "urls_pdf", collapse = FALSE)
```

Arguments

record_list	list with one record of create_endnote_list()
col_name	default: "url_pdfurls"
collapse	should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row pdfurls data frame

get_reference_type_names

Helper function: get reference type names

Description

Helper function: get reference type names

Usage

```
get_reference_type_names(endnote_xml = default_xml())
```

Arguments

endnote_xml path to Endnote library exported as .xml (default: default_xml())

Value

data.frame with columns record_id, rec_number, ref_type_id, ref_type_name

Examples

```
ref_type_names <- get_reference_type_names()
head(ref_type_names)
```

get_secondary_authors *Helper function: get secondary authors from list for a reference*

Description

Helper function: get secondary authors from list for a reference

Usage

```
get_secondary_authors(record_list, collapse = FALSE)
```

Arguments

record_list list with one record of create_endnote_list()
collapse should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row authors data frame

get_tertiary_authors *Helper function: get tertiary authors from list for a reference*

Description

Helper function: get tertiary authors from list for a reference

Usage

```
get_tertiary_authors(record_list, collapse = FALSE)
```

Arguments

record_list list with one record of create_endnote_list()
collapse should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

one row authors data frame

get_xml_filename_without_extension
Helper function: get xml filename without extension

Description

Helper function: get xml filename without extension

Usage

```
get_xml_filename_without_extension(obj)
```

Arguments

obj list or dataframe as retrieved by create_endnote_list() or create_df_from_endnote_xml()

Value

xml filename without file extension

`give_hints_accessibility`*Helper Function: Give Hints For Accessibility*

Description

Helper Function: Give Hints For Accessibility

Usage

```
give_hints_accessibility(access, dbg = TRUE)
```

Arguments

<code>access</code>	vector with accessibility metadata to check
<code>dbg</code>	should debug messages be printed (default: TRUE)

Value

vector with access info with hints how to improve data quality (in case `give_hints = TRUE`)

`give_hints_author_names`*Helper Function: Give Hints For Author Names*

Description

Helper Function: Give Hints For Author Names

Usage

```
give_hints_author_names(author_names, dbg = TRUE)
```

Arguments

<code>author_names</code>	vector with author names to check
<code>dbg</code>	should debug messages be printed (default: TRUE)

Value

vector with `author_names` with hints how to improve data quality (in case `give_hints = TRUE`)

give_hints_project_names

Helper Function: Give Hints For Project Names

Description

Helper Function: Give Hints For Project Names

Usage

```
give_hints_project_names(project_names, dbg = TRUE)
```

Arguments

project_names vector with project names to check
dbg should debug messages be printed (default: TRUE)

Value

vector with project_names with hints how to improve data quality (in case give_hints = TRUE)

plot_pubs_by_author *Plot Number of Publications by Author*

Description

Plot Number of Publications by Author

Usage

```
plot_pubs_by_author(pubs_by_author_df)
```

Arguments

pubs_by_author_df
 a data frame with author names in column "value" and number of publications
 (in column "n")

Value

plot of number of publications per author

Examples

```
refs_by_author_lastfirst <- create_df_from_endnote_xml() %>%
  dplyr::filter(.data$key2 == "authors") %>%
  dplyr::count(.data$value) %>%
  dplyr::arrange(dplyr::desc(.data$n))
plot_pubs_by_author(refs_by_author_lastfirst[1:30, ])
```

plot_pubs_by_year *Plot Publications By Year*

Description

Plot Publications By Year

Usage

```
plot_pubs_by_year(refs_df)
```

Arguments

refs_df reference dataframe as retrieved by `kwb.endnote::create_references_df()`

Value

plot with publications by year

Examples

```
endnote_list <- kwb.endnote::create_endnote_list()
refs_df <- kwb.endnote::create_references_df(endnote_list)
plot_pubs_by_year(refs_df)
```

plot_wordcloud_keywords *Plot Wordcloud Keywords*

Description

Plot Wordcloud Keywords

Usage

```
plot_wordcloud_keywords(keywords_df, ...)
```

Arguments

keywords_df keywords dataframe as retrieved by create_keywords_df()
 ... additional arguments passed to wordcloud2::wordcloud2()

Value

wordcloud keywords plot

record_list_to_df *Reference List to Data Frame*

Description

Reference List to Data Frame

Usage

```
record_list_to_df(record_list, collapse = FALSE)
```

Arguments

record_list list with one record of create_endnote_list()
 collapse should separate fields in "style" be collapsed to one field? (default: FALSE)

Value

data frame for record

tidy_df *Helper function: tidy dataframe*

Description

Helper function: tidy dataframe

Usage

```
tidy_df(df, exclude_cols = "rec_number")
```

Arguments

df data frame as retrieved by create_references_df() or clean_references_df()
 exclude_cols vector of column names to exclude for gathering (default: "rec_number")

Value

a tidy dataframe with columns rec_number, key and value

Examples

```
## Not run:  
endnote_list <- create_endnote_list()  
refs_df <- create_references_df(endnote_list)  
refs_df_tidy <- tidy_df(refs_df)  
  
## End(Not run)
```

tidy_multi_cols_df *Helper Function: Tidy Multi Cols Dataframe*

Description

Helper Function: Tidy Multi Cols Dataframe

Usage

```
tidy_multi_cols_df(df)
```

Arguments

df as retrieved by create_references_df() or clean_references_df()

Value

tidy dataframe for all multi cols in df

Examples

```
## Not run: endnote_list <- create_endnote_list()  
refs_clean_df <- clean_references_df(endnote_list)  
multi_cols_df <- tidy_multi_cols_df(refs_clean_df)  
  
## End(Not run)
```

tidy_multi_cols_list *Helper Function: Tidy Multi Cols List*

Description

Helper Function: Tidy Multi Cols List

Usage

```
tidy_multi_cols_list(df)
```

Arguments

df as retrieved by create_references_df() or clean_references_df()

Value

tidy list with a sublist for each multi col in df (see: get_available_multi_cols())

Examples

```
## Not run: endnote_list <- create_endnote_list()
refs_clean_df <- clean_references_df(endnote_list)
multi_cols_list <- tidy_multi_cols_list(refs_clean_df)

## End(Not run)
```

tidy_selected_cols *Helper Function: Tidy Selected Cols*

Description

Helper Function: Tidy Selected Cols

Usage

```
tidy_selected_cols(df, column = "author")
```

Arguments

df as retrieved by create_references_df() or clean_references_df()
column a multi col column e.g. "author"

Value

tidy dataframe for selected multi col in df

```
write_clean_references_df_to_xlsx
      Write Clean References Dataframe to XLSX
```

Description

Write Clean References Dataframe to XLSX

Usage

```
write_clean_references_df_to_xlsx(
  endnote_list,
  file = default_clean_xlsx(endnote_list),
  export_dir = ".",
  give_hints = FALSE,
  dbg = TRUE,
  ...
)
```

Arguments

endnote_list	list created with create_endnote_list()
file	name of file to save (default: default_clean_xlsx(endnote_list))
export_dir	directory where to save 'file' (default: ".")
give_hints	if TRUE hints will be generated, e.g. "add_public_or_confidential" for accessibility data
dbg	show debug messages (default: TRUE)
...	additional arguments passed to openxlsx::write.xlsx() publication type

Examples

```
## Not run: endnote_list <- create_endnote_list()
write_clean_references_df_to_xlsx(endnote_list)

## End(Not run)
```

```
write_references_df_to_xlsx
      Write References Dataframe to XLSX
```

Description

Write References Dataframe to XLSX

Usage

```
write_references_df_to_xlsx(  
  endnote_list,  
  file = default_xlsx(endnote_list),  
  export_dir = ".",  
  dbg = TRUE,  
  ...  
)
```

Arguments

endnote_list	list created with create_endnote_list()
file	name of file to save (default: default_xlsx(endnote_list))
export_dir	directory where to save 'file' (default: ".")
dbg	show debug messages (default: TRUE)
...	additional arguments passed to openxlsx::write.xlsx()

Value

write references dataframe to xlsx with one sheet for each publication type

Examples

```
## Not run: endnote_list <- create_endnote_list()  
write_references_df_to_xlsx(endnote_list)  
  
## End(Not run)
```


Index

[add_file_info_attributes](#), 2

[check_for_differences](#), 3

[check_problematic_entries](#), 4

[clean_accessibility](#), 5

[clean_author_names](#), 5

[clean_dois](#), 6

[clean_project_names](#), 6

[clean_references_df](#), 7

[create_df_from_endnote_xml](#), 7

[create_endnote_list](#), 8

[create_keywords_df](#), 8

[create_list_by_pubtype_from_df](#), 9

[create_list_with_unique_entries](#), 10

[create_references_df](#), 10

[default_clean_xlsx](#), 11

[default_xlsx](#), 11

[default_xml](#), 12

[extdata_file](#), 12

[get_abstract](#), 12

[get_authors](#), 13

[get_available_multi_cols](#), 13

[get_keywords](#), 14

[get_pdfurls](#), 14

[get_reference_type_names](#), 15

[get_secondary_authors](#), 15

[get_tertiary_authors](#), 16

[get_xml_filename_without_extension](#), 16

[give_hints_accessibility](#), 17

[give_hints_author_names](#), 17

[give_hints_project_names](#), 18

[plot_pubs_by_author](#), 18

[plot_pubs_by_year](#), 19

[plot_wordcloud_keywords](#), 19

[record_list_to_df](#), 20

[system.file](#), 12

[tidy_df](#), 20

[tidy_multi_cols_df](#), 21

[tidy_multi_cols_list](#), 22

[tidy_selected_cols](#), 22

[write_clean_references_df_to_xlsx](#), 23

[write_references_df_to_xlsx](#), 23