

Package ‘ready4fun’

July 31, 2020

Title Readyforwhatsnext Function Authoring And Documentation Tools

Version 0.0.0.9088

Description ready4fun is a collection of functions for authoring code libraries of functions and datasets for use in mental health simulations developed within the readyforwhatsnext ecosystem.

Depends R (>= 3.5.1)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

Imports Hmisc,
devtools,
dplyr,
magrittr,
methods,
purrr,
readxl,
rlang,
sinew,
stats,
stringi,
stringr,
testit,
tibble,
usethis

Suggests knitr,
rmarkdown

R topics documented:

ready4fun-package	3
abbreviations_lup	4
add_indefartls_to_phrases_chr_vec	4
add_indef_artl_to_item_chr_vec	5
add_plurals_to_abbr_lup_tb	6

assert_does_not_match_terms	6
close_open_sinks	7
fn_type_lup_tb	7
force_req_pkg_install	8
get_fn_args_chr_vec	8
get_fn_nms_in_file_chr	9
get_from_lup_obj	9
get_outp_obj_type_chr_vec	10
get_r4_obj_slots_chr_vec	10
get_return_obj_nm_chr	11
import_xls_sheets_ls	11
make_abbr_lup_tb	12
make_and_doc_fn_type_R	13
make_arg_desc_chr_vec	13
make_arg_desc_ls	14
make_arg_desc_spine_chr	14
make_arg_title_chr_vec	15
make_arg_type_abbr_chr_vec	16
make_arg_type_abbr_spine_chr	16
make_arg_type_chr_vec	17
make_arg_type_lup_ls	17
make_fn_desc_chr_vec	18
make_fn_desc_spine_chr_vec	18
make_fn_dmt_spine_chr_ls	19
make_fn_dmt_tbl_tb	20
make_fn_dmt_tbl_tpl_tb	20
make_fn_title_chr_vec	21
make_fn_type_lup_tb	21
make_gtr_str_dmt_spine_chr_ls	22
make_new_fn_dmt_chr_ls	22
make_obj_lup_tb	23
make_std_fn_dmt_spine_chr_ls	24
object_type_lup	25
read_fns	25
remove_collate_chr_vec	26
remove_obj_type_from_nm_chr_vec	26
replace_abbr_chr	27
reset_pkg_files_R	27
rowbind_all_tbs_in_r4_obj_r4	28
rowbind_tbs_in_r4_obj_r4	28
unload_packages	29
update_abbr_lup_tb	29
update_first_word_case_chr	30
update_fns_dmt_tb_chr_vars_chr_vec	31
update_fns_dmt_tb_lgl_vars_chr_vec	31
update_fns_dmt_tb_ls_vars_chr_vec	32
update_fns_dmt_tb_tb	32
update_fn_dmt_chr	33
update_fn_dmt_with_slots_chr	34
update_ns_chr	35
write_all_tbs_in_tbs_r4_to_csvs	35
write_and_doc_ds_R	36

write_and_doc_fn_fls_R	37
write_documented_fns_R	37
write_ds_dmt_R	38
write_fn_dmt	39
write_fn_fl_R	40
write_from_tmp_R	40
write_nsimps_to_desc	41
write_pkg_R	41
write_pt_lup_db_R	42
write_std_imp_R	42
write_tb_to_csv	43
write_ws	43
%>%	43
Index	44

ready4fun-package	<i>ready4fun: Readyforwhatsnext Function Authoring And Documenta- tion Tools</i>
-------------------	--

Description

ready4fun is a collection of functions for authoring code libraries of functions and datasets for use in mental health simulations developed within the readyforwhatsnext ecosystem.

Details

To learn more about ready4fun, start with the vignettes: ‘browseVignettes(package = "ready4fun")’

Second section

Some text.

Author(s)

Maintainer: Matthew Hamilton <matthew.hamilton@orygen.org.au> ([ORCID](#))

Authors:

- Glen Wiesner <Glen.Wiesner@vu.edu.au> ([ORCID](#))

Other contributors:

- Orygen [copyright holder, funder]
- VicHealth [funder]
- Victoria University [funder]

abbreviations_lup	<i>Common abbreviations lookup table</i>
-------------------	--

Description

A lookup table for abbreviations commonly used in object names in the ready4funpackage.

Usage

```
abbreviations_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 173 rows and 3 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

plural_lgl Plural (a logical vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

add_indefartls_to_phrases_chr_vec	<i>Add indefinite articles to phrases</i>
-----------------------------------	---

Description

`add_indefartls_to_phrases_chr_vec()` is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add indefinite articles to phrases. Function argument `abbreviated_phrase_chr_vec` specifies the object to be updated. The function returns phrases (a character vector).

Usage

```
add_indefartls_to_phrases_chr_vec(
  abbreviated_phrase_chr_vec,
  abbreviations_lup = NULL,
  ignore_phrs_not_in_lup_lgl = T
)
```

Arguments

abbreviated_phrase_chr_vec
Abbreviated phrase (a character vector)

abbreviations_lup
Abbreviations (a lookup table), Default: NULL

ignore_phrs_not_in_lup_lgl
Ignore phrases not in lookup table (a logical vector of length 1), Default: T

Value

Phrases (a character vector)

add_indef_artl_to_item_chr_vec
Add indefinite article to item

Description

add_indef_artl_to_item_chr_vec() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add an indefinite article to item. Function argument phrase_chr_vec specifies the object to be updated. The function returns an indefinite item (a character vector).

Usage

```
add_indef_artl_to_item_chr_vec(
  phrase_chr_vec,
  abbreviations_lup = NULL,
  ignore_phrs_not_in_lup_lgl = T
)
```

Arguments

phrase_chr_vec Phrase (a character vector)

abbreviations_lup
Abbreviations (a lookup table), Default: NULL

ignore_phrs_not_in_lup_lgl
Ignore phrases not in lookup table (a logical vector of length 1), Default: T

Value

Indefinite item (a character vector)

`add_plurals_to_abbr_lup_tb`*Add plurals to abbreviation lookup table*

Description

`add_plurals_to_abbr_lup_tb()` is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add plurals to an abbreviation lookup table. Function argument `abbr_tb` specifies the object to be updated. The function returns an abbreviation (a tibble).

Usage

```
add_plurals_to_abbr_lup_tb(  
  abbr_tb,  
  no_plural_chr_vec = NA_character_,  
  custom_plural_ls = NULL  
)
```

Arguments

<code>abbr_tb</code>	Abbreviation (a tibble)
<code>no_plural_chr_vec</code>	No plural (a character vector), Default: 'NA'
<code>custom_plural_ls</code>	Custom plural (a list), Default: NULL

Value

Abbreviation (a tibble)

`assert_does_not_match_terms`*Assert does not match terms*

Description

`assert_does_not_match_terms()` is an Assert function that validates that an object conforms to required condition(s). If the object does not meet all required conditions, program execution will be stopped and an error message provided. Specifically, this function implements an algorithm to assert does not match terms. Function argument `input_chr_vec` specifies the object on which assert validation checks are to be performed. Argument `exclude_if_match_chr_vec` provides the object containing values used for validation tests. The function is called for its side effects and does not return a value.

Usage

```
assert_does_not_match_terms(input_chr_vec, exclude_if_match_chr_vec)
```

Arguments

input_chr_vec Input (a character vector)
exclude_if_match_chr_vec Exclude if match (a character vector)

close_open_sinks	<i>Close open sinks</i>
------------------	-------------------------

Description

close_open_sinks() is a Close function that closes specified connections. Specifically, this function implements an algorithm to close open sinks. The function is called for its side effects and does not return a value.

Usage

close_open_sinks()

fn_type_lup_tb	<i>Function type lookup table</i>
----------------	-----------------------------------

Description

A lookup table to find descriptions for different types of functions used within the ready4funpackage suite.

Usage

fn_type_lup_tb

Format

An object of class tbl_df (inherits from tbl, data.frame) with 16 rows and 4 columns.

Details

A tibble

fn_type_nm_chr Function type name (a character vector)
fn_type_desc_chr Function type description (a character vector)
first_arg_desc_chr First argument description (a character vector)
second_arg_desc_chr Second argument description (a character vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

force_req_pkg_install	<i>Force req package install</i>
-----------------------	----------------------------------

Description

force_req_pkg_install() is a Force function that checks if a specified local or global environmental condition is met and if not, updates the specified environment to comply with the condition. Specifically, this function implements an algorithm to force req a package install. The function is called for its side effects and does not return a value.

Usage

```
force_req_pkg_install(package_nm_chr)
```

Arguments

package_nm_chr Package name (a character vector of length 1)

get_fn_args_chr_vec	<i>Get function arguments</i>
---------------------	-------------------------------

Description

get_fn_args_chr_vec() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get a function arguments. Function argument fn specifies the where to look for the required object. The function returns a function arguments (a character vector).

Usage

```
get_fn_args_chr_vec(fn)
```

Arguments

fn Function (a function)

Value

Function arguments (a character vector)

get_fn_nms_in_file_chr

Get function names in file

Description

get_fn_nms_in_file_chr() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get a function names in a file. Function argument path_chr specifies the where to look for the required object. The function returns local (a character vector of length 1).

Usage

```
get_fn_nms_in_file_chr(path_chr)
```

Arguments

path_chr Path (a character vector of length 1)

Value

Local (a character vector of length 1)

get_from_lup_obj

Get from lookup table object

Description

get_from_lup_obj() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get from a lookup table object. Function argument data_lookup_tb specifies the where to look for the required object. The function returns return object (an output object of multiple potential types).

Usage

```
get_from_lup_obj(
  data_lookup_tb,
  match_value_xx,
  match_var_nm_chr,
  target_var_nm_chr,
  evaluate_lgl = TRUE
)
```

Arguments

data_lookup_tb Data lookup (a tibble)
 match_value_xx Match value (an output object of multiple potential types)
 match_var_nm_chr Match var name (a character vector of length 1)
 target_var_nm_chr Target var name (a character vector of length 1)
 evaluate_lgl Evaluate (a logical vector of length 1), Default: TRUE

Value

Return object (an output object of multiple potential types)

get_outp_obj_type_chr_vec
Get output object type

Description

get_outp_obj_type_chr_vec() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get an output object type. Function argument fns_chr_vec specifies the where to look for the required object. The function returns an output object type (a character vector).

Usage

```
get_outp_obj_type_chr_vec(fns_chr_vec)
```

Arguments

fns_chr_vec Functions (a character vector)

Value

Output object type (a character vector)

get_r4_obj_slots_chr_vec
Get readyforwhatsnext S4 object slots

Description

get_r4_obj_slots_chr_vec() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get a readyforwhatsnext S4 object slots. Function argument fn_name_chr specifies the where to look for the required object. The function returns slots (a character vector).

Usage

```
get_r4_obj_slots_chr_vec(fn_name_chr, package_chr = "")
```

Arguments

fn_name_chr Function name (a character vector of length 1)
 package_chr Package (a character vector of length 1), Default: ""

Value

Slots (a character vector)

get_return_obj_nm_chr *Get return object name*

Description

get_return_obj_nm_chr() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get return an object name. Function argument fn specifies the where to look for the required object. The function returns return (a character vector of length 1).

Usage

```
get_return_obj_nm_chr(fn)
```

Arguments

fn Function (a function)

Value

Return (a character vector of length 1)

import_xls_sheets_ls *Import Excel workbook sheets*

Description

import_xls_sheets_ls() is an Import function that reads a data object in its native format and converts it to an R object. Specifically, this function implements an algorithm to an import Excel workbook sheets. The function returns a tibble list (a list of tibbles).

Usage

```
import_xls_sheets_ls(range_chr, sheet_names_chr_vec, path_chr)
```

Arguments

range_chr Range (a character vector of length 1)
 sheet_names_chr_vec Sheet names (a character vector)
 path_chr Path (a character vector of length 1)

Value

Tibble list (a list of tibbles)

make_abbr_lup_tb	<i>Make abbreviation lookup table</i>
------------------	---------------------------------------

Description

make_abbr_lup_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an abbreviation lookup table. The function is called for its side effects and does not return a value.

Usage

```
make_abbr_lup_tb(
  short_name_chr_vec = NA_character_,
  long_name_chr_vec = NA_character_,
  no_plural_chr_vec = NA_character_,
  custom_plural_ls = NULL,
  overwrite_lgl = T,
  url_chr,
  pkg_nm_chr
)
```

Arguments

short_name_chr_vec Short name (a character vector), Default: 'NA'
 long_name_chr_vec Long name (a character vector), Default: 'NA'
 no_plural_chr_vec No plural (a character vector), Default: 'NA'
 custom_plural_ls Custom plural (a list), Default: NULL
 overwrite_lgl Overwrite (a logical vector of length 1), Default: T
 url_chr Url (a character vector of length 1)
 pkg_nm_chr Package name (a character vector of length 1)

make_and_doc_fn_type_R

Make and document function type

Description

make_and_doc_fn_type_R() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make and a document function type R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
make_and_doc_fn_type_R(
  fn_type_lup_tb = make_fn_type_lup_tb(),
  overwrite_lgl = T,
  pkg_nm_chr,
  url_chr = url_chr
)
```

Arguments

fn_type_lup_tb	Function type lookup table (a tibble), Default: make_fn_type_lup_tb()
overwrite_lgl	Overwrite (a logical vector of length 1), Default: T
pkg_nm_chr	Package name (a character vector of length 1)
url_chr	Url (a character vector of length 1), Default: url_chr

make_arg_desc_chr_vec *Make argument description*

Description

make_arg_desc_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument description. The function returns an argument description (a character vector).

Usage

```
make_arg_desc_chr_vec(
  fn_args_chr_vec,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

Arguments

fn_args_chr_vec
Function arguments (a character vector)

object_type_lup
Object type (a lookup table), Default: NULL

abbreviations_lup
Abbreviations (a lookup table), Default: NULL

Value

Argument description (a character vector)

make_arg_desc_ls	<i>Make argument description</i>
------------------	----------------------------------

Description

make_arg_desc_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument description. The function is called for its side effects and does not return a value.

Usage

```
make_arg_desc_ls(fn_nms_chr_vec)
```

Arguments

fn_nms_chr_vec Function names (a character vector)

make_arg_desc_spine_chr	<i>Make argument description spine</i>
-------------------------	--

Description

make_arg_desc_spine_chr() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument description spine. The function returns an argument description spine (a character vector of length 1).

Usage

```
make_arg_desc_spine_chr(  
  argument_nm_chr,  
  object_type_lup = NULL,  
  abbreviations_lup = NULL  
)
```

Arguments

argument_nm_chr
Argument name (a character vector of length 1)

object_type_lup
Object type (a lookup table), Default: NULL

abbreviations_lup
Abbreviations (a lookup table), Default: NULL

Value

Argument description spine (a character vector of length 1)

make_arg_title_chr_vec
Make argument title

Description

make_arg_title_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument title. The function returns title (a character vector).

Usage

```
make_arg_title_chr_vec(
  args_chr_vec,
  match_chr_vec,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

Arguments

args_chr_vec Arguments (a character vector)

match_chr_vec Match (a character vector)

object_type_lup
Object type (a lookup table), Default: NULL

abbreviations_lup
Abbreviations (a lookup table), Default: NULL

Value

Title (a character vector)

```
make_arg_type_abbr_chr_vec
```

Make argument type abbreviation

Description

make_arg_type_abbr_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument type an abbreviation. The function returns an argument type abbreviation (a character vector).

Usage

```
make_arg_type_abbr_chr_vec(fn_args_chr_vec, object_type_lup = NULL)
```

Arguments

```
fn_args_chr_vec
```

Function arguments (a character vector)

```
object_type_lup
```

Object type (a lookup table), Default: NULL

Value

Argument type abbreviation (a character vector)

```
make_arg_type_abbr_spine_chr
```

Make argument type abbreviation spine

Description

make_arg_type_abbr_spine_chr() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument type an abbreviation spine. The function returns an argument type abbreviation spine (a character vector of length 1).

Usage

```
make_arg_type_abbr_spine_chr(argument_nm_chr, lup_tb)
```

Arguments

```
argument_nm_chr
```

Argument name (a character vector of length 1)

```
lup_tb
```

Lookup table (a tibble)

Value

Argument type abbreviation spine (a character vector of length 1)

make_arg_type_chr_vec *Make argument type*

Description

make_arg_type_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument type. The function returns an argument description (a character vector).

Usage

```
make_arg_type_chr_vec(  
  fn_args_chr_vec,  
  object_type_lup = NULL,  
  abbreviations_lup = NULL,  
  fn  
)
```

Arguments

fn_args_chr_vec	Function arguments (a character vector)
object_type_lup	Object type (a lookup table), Default: NULL
abbreviations_lup	Abbreviations (a lookup table), Default: NULL
fn	Function (a function)

Value

Argument description (a character vector)

make_arg_type_lup_ls *Make argument type*

Description

make_arg_type_lup_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an argument type. The function returns a lookup table list (a list of lookup tables).

Usage

```
make_arg_type_lup_ls(object_type_lup = NULL)
```

Arguments

object_type_lup	Object type (a lookup table), Default: NULL
-----------------	---

Value

Lookup table list (a list of lookup tables)

make_fn_desc_chr_vec *Make function description*

Description

make_fn_desc_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function description. The function returns a function description (a character vector).

Usage

```
make_fn_desc_chr_vec(fns_chr_vec, title_chr_vec, output_chr_vec)
```

Arguments

fns_chr_vec Functions (a character vector)
 title_chr_vec Title (a character vector)
 output_chr_vec Output (a character vector)

Value

Function description (a character vector)

make_fn_desc_spine_chr_vec
 Make function description spine

Description

make_fn_desc_spine_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function description spine. The function returns a function description spine (a character vector).

Usage

```
make_fn_desc_spine_chr_vec(fn_name_chr, fn_title_chr, fn_type_lup_tb = NULL)
```

Arguments

fn_name_chr Function name (a character vector of length 1)
 fn_title_chr Function title (a character vector of length 1)
 fn_type_lup_tb Function type lookup table (a tibble), Default: NULL

Value

Function description spine (a character vector)

`make_fn_dmt_spine_chr_ls`*Make function documentation spine*

Description

`make_fn_dmt_spine_chr_ls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function documentation spine. The function returns a function documentation spine (a list of character vectors of length 1).

Usage

```
make_fn_dmt_spine_chr_ls(  
  fn_name_chr,  
  fn_type_chr,  
  fn_title_chr = NA_character_,  
  fn,  
  details_chr = NA_character_,  
  example_lgl = F,  
  export_lgl = T,  
  class_name_chr,  
  doc_in_class_lgl  
)
```

Arguments

<code>fn_name_chr</code>	Function name (a character vector of length 1)
<code>fn_type_chr</code>	Function type (a character vector of length 1)
<code>fn_title_chr</code>	Function title (a character vector of length 1), Default: 'NA'
<code>fn</code>	Function (a function)
<code>details_chr</code>	Details (a character vector of length 1), Default: 'NA'
<code>example_lgl</code>	Example (a logical vector of length 1), Default: F
<code>export_lgl</code>	Export (a logical vector of length 1), Default: T
<code>class_name_chr</code>	Class name (a character vector of length 1)
<code>doc_in_class_lgl</code>	Document in class (a logical vector of length 1)

Value

Function documentation spine (a list of character vectors of length 1)

make_fn_dmt_tbl_tb	<i>Make function documentation table</i>
--------------------	--

Description

make_fn_dmt_tbl_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function documentation a table. The function returns a function documentation table (a tibble).

Usage

```
make_fn_dmt_tbl_tb(
  fns_path_chr_vec,
  fns_dir_chr,
  pkg_nm_chr,
  custom_dmt_ls = list(title_ls = NULL, desc_ls = NULL, details_ls = NULL, export_ls =
    NULL, output_ls = NULL, example_ls = NULL, args_ls_ls = NULL),
  append_lgl = T
)
```

Arguments

fns_path_chr_vec	Functions path (a character vector)
fns_dir_chr	Functions directory (a character vector of length 1)
pkg_nm_chr	Package name (a character vector of length 1)
custom_dmt_ls	Custom documentation (a list), Default: list(title_ls = NULL, desc_ls = NULL, details_ls = NULL, export_ls = NULL, output_ls = NULL, example_ls = NULL, args_ls_ls = NULL)
append_lgl	Append (a logical vector of length 1), Default: T

Value

Function documentation table (a tibble)

make_fn_dmt_tbl_tpl_tb	<i>Make function documentation table template</i>
------------------------	---

Description

make_fn_dmt_tbl_tpl_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function documentation a table template. The function returns a function documentation table (a tibble).

Usage

```
make_fn_dmt_tbl_tpl_tb(fns_path_chr_vec, fns_dir_chr)
```

Arguments

fns_path_chr_vec Functions path (a character vector)

fns_dir_chr Functions directory (a character vector of length 1)

Value

Function documentation table (a tibble)

make_fn_title_chr_vec *Make function title*

Description

make_fn_title_chr_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function title. The function returns title (a character vector).

Usage

```
make_fn_title_chr_vec(
  fns_chr_vec,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

Arguments

fns_chr_vec Functions (a character vector)

object_type_lup Object type (a lookup table), Default: NULL

abbreviations_lup Abbreviations (a lookup table), Default: NULL

Value

Title (a character vector)

make_fn_type_lup_tb *Make function type lookup table*

Description

make_fn_type_lup_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a function type a lookup table. The function returns a function type lookup table (a tibble).

Usage

```
make_fn_type_lup_tb()
```

Value

Function type lookup table (a tibble)

```
make_gtr_str_dmt_spine_chr_ls
```

Make getter setter documentation spine

Description

`make_gtr_str_dmt_spine_chr_ls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a getter setter documentation spine. The function returns a getter setter documentation spine (a list of character vectors of length 1).

Usage

```
make_gtr_str_dmt_spine_chr_ls(
  fn_type_chr,
  fn_name_chr,
  class_name_chr,
  doc_in_class_lgl,
  example_lgl = F
)
```

Arguments

<code>fn_type_chr</code>	Function type (a character vector of length 1)
<code>fn_name_chr</code>	Function name (a character vector of length 1)
<code>class_name_chr</code>	Class name (a character vector of length 1)
<code>doc_in_class_lgl</code>	Document in class (a logical vector of length 1)
<code>example_lgl</code>	Example (a logical vector of length 1), Default: F

Value

Getter setter documentation spine (a list of character vectors of length 1)

```
make_new_fn_dmt_chr_ls
```

Make new function documentation

Description

`make_new_fn_dmt_chr_ls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make new a function documentation. The function returns new function documentation (a list of character vectors of length 1).

Usage

```
make_new_fn_dmt_chr_ls(
  fn_type_chr,
  fn_name_chr,
  fn_desc_chr = NA_character_,
  fn_det_chr = NA_character_,
  fn_out_type_chr = NA_character_,
  args_ls = NULL,
  fn = NULL,
  object_type_lup = NULL
)
```

Arguments

fn_type_chr	Function type (a character vector of length 1)
fn_name_chr	Function name (a character vector of length 1)
fn_desc_chr	Function description (a character vector of length 1), Default: 'NA'
fn_det_chr	Function det (a character vector of length 1), Default: 'NA'
fn_out_type_chr	Function out type (a character vector of length 1), Default: 'NA'
args_ls	Arguments (a list), Default: NULL
fn	Function (a function), Default: NULL
object_type_lup	Object type (a lookup table), Default: NULL

Value

New function documentation (a list of character vectors of length 1)

make_obj_lup_tb	<i>Make object lookup table</i>
-----------------	---------------------------------

Description

make_obj_lup_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make an object lookup table. The function returns an object (a tibble).

Usage

```
make_obj_lup_tb()
```

Value

Object (a tibble)

make_std_fn_dmt_spine_chr_ls

Make standard function documentation spine

Description

make_std_fn_dmt_spine_chr_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make a standard function documentation spine. The function returns a standard function documentation spine (a list of character vectors of length 1).

Usage

```
make_std_fn_dmt_spine_chr_ls(
  fn_name_chr,
  fn_type_chr,
  fn_title_chr,
  fn,
  details_chr = NA_character_,
  example_lgl = F,
  export_lgl = T,
  class_name_chr = "",
  exclude_if_match_chr_vec
)
```

Arguments

fn_name_chr	Function name (a character vector of length 1)
fn_type_chr	Function type (a character vector of length 1)
fn_title_chr	Function title (a character vector of length 1)
fn	Function (a function)
details_chr	Details (a character vector of length 1), Default: 'NA'
example_lgl	Example (a logical vector of length 1), Default: F
export_lgl	Export (a logical vector of length 1), Default: T
class_name_chr	Class name (a character vector of length 1), Default: ''
exclude_if_match_chr_vec	Exclude if match (a character vector)

Value

Standard function documentation spine (a list of character vectors of length 1)

object_type_lup	<i>Object abbreviations lookup table</i>
-----------------	--

Description

A lookup table to identify R object types from an abbreviation that can be used as object name suffices.

Usage

```
object_type_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 100 rows and 3 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

plural_lgl Plural (a logical vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

read_fns	<i>Read functions</i>
----------	-----------------------

Description

`read_fns()` is a Read function that reads an R script into memory. Specifically, this function implements an algorithm to read a functions. Function argument `fns_dir_chr` specifies the path to object. The function returns a functions path (a character vector).

Usage

```
read_fns(fns_dir_chr = "data-raw/fns/")
```

Arguments

fns_dir_chr Functions directory (a character vector of length 1), Default: `'data-raw/fns/'`

Value

Functions path (a character vector)

```
remove_collate_chr_vec
```

Remove collate

Description

remove_collate_chr_vec() is a Remove function that edits an object, removing a specified element or elements. Specifically, this function implements an algorithm to remove collate. Function argument description_chr specifies the object to be updated. Argument NA provides the object to be updated. The function returns a description (a character vector of length 1).

Usage

```
remove_collate_chr_vec(description_chr)
```

Arguments

```
description_chr
```

Description (a character vector of length 1)

Value

Description (a character vector of length 1)

```
remove_obj_type_from_nm_chr_vec
```

Remove object type from name

Description

remove_obj_type_from_nm_chr_vec() is a Remove function that edits an object, removing a specified element or elements. Specifically, this function implements an algorithm to remove an object type from a name. Function argument nms_chr_vec specifies the object to be updated. Argument object_type_lup provides the object to be updated. The function returns names (a character vector).

Usage

```
remove_obj_type_from_nm_chr_vec(
  nms_chr_vec,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

Arguments

```
nms_chr_vec
```

Names (a character vector)

```
object_type_lup
```

Object type (a lookup table), Default: NULL

```
abbreviations_lup
```

Abbreviations (a lookup table), Default: NULL

Value

Names (a character vector)

replace_abbr_chr	<i>Replace abbreviation</i>
------------------	-----------------------------

Description

replace_abbr_chr() is a Replace function that edits an object, replacing a specified element with another specified element. Specifically, this function implements an algorithm to replace an abbreviation. Function argument title_chr specifies the object to be updated. Argument abbreviations_lup provides the object to be updated. The function returns title (a character vector of length 1).

Usage

```
replace_abbr_chr(title_chr, abbreviations_lup = NULL, collapse_lgl = T)
```

Arguments

title_chr Title (a character vector of length 1)
 abbreviations_lup Abbreviations (a lookup table), Default: NULL
 collapse_lgl Collapse (a logical vector of length 1), Default: T

Value

Title (a character vector of length 1)

reset_pkg_files_R	<i>Reset package files</i>
-------------------	----------------------------

Description

reset_pkg_files_R() is a Reset function that edits an object, overwriting the current version with a default version. Specifically, this function implements an algorithm to reset a package files R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
reset_pkg_files_R(  
  package_chr,  
  package_dir = getwd(),  
  description_ls = NULL,  
  keep_version_lgl = T  
)
```

Arguments

package_chr Package (a character vector of length 1)
 package_dir PARAM_DESCRIPTION, Default: getwd()
 description_ls Description (a list), Default: NULL
 keep_version_lgl Keep version (a logical vector of length 1), Default: T

rowbind_all_tbs_in_r4_obj_r4

Rowbind all tibbles in readyforwhatsnext S4 object

Description

rowbind_all_tbs_in_r4_obj_r4() is a Rowbind function that performs custom rowbind operations on table objects. Specifically, this function implements an algorithm to rowbind all a tibbles in a readyforwhatsnext S4 object. The function returns a tibbles (a readyforwhatsnext s4).

Usage

```
rowbind_all_tbs_in_r4_obj_r4(tbs_r4, second_tbs_r4, r4_name_chr)
```

Arguments

tbs_r4 Tibbles (a readyforwhatsnext S4)
 second_tbs_r4 Second tibbles (a readyforwhatsnext S4)
 r4_name_chr Readyforwhatsnext S4 name (a character vector of length 1)

Value

Tibbles (a readyforwhatsnext S4)

rowbind_tbs_in_r4_obj_r4

Rowbind tibbles in readyforwhatsnext S4 object

Description

rowbind_tbs_in_r4_obj_r4() is a Rowbind function that performs custom rowbind operations on table objects. Specifically, this function implements an algorithm to rowbind a tibbles in a readyforwhatsnext S4 object. The function returns a tibbles (a readyforwhatsnext s4).

Usage

```
rowbind_tbs_in_r4_obj_r4(tbs_r4, slot_nm_chr, second_tbs_r4, r4_name_chr)
```

Arguments

tbs_r4	Tibbles (a readyforwhatsnext S4)
slot_nm_chr	Slot name (a character vector of length 1)
second_tbs_r4	Second tibbles (a readyforwhatsnext S4)
r4_name_chr	Readyforwhatsnext S4 name (a character vector of length 1)

Value

Tibbles (a readyforwhatsnext S4)

unload_packages	<i>Unload packages</i>
-----------------	------------------------

Description

unload_packages() is an Unload function that performs a custom detaching of a package from the search path. Specifically, this function implements an algorithm to unload packages. Function argument package_chr_vec specifies the package(s) to be detached from the search path. Argument NA provides the package(s) to be detached from the search path. The function is called for its side effects and does not return a value.

Usage

```
unload_packages(package_chr_vec)
```

Arguments

package_chr_vec	Package (a character vector)
-----------------	------------------------------

update_abbr_lup_tb	<i>Update abbreviation lookup table</i>
--------------------	---

Description

update_abbr_lup_tb() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update abbreviation lookup table. Function argument abbr_tb specifies the object to be updated. Argument short_name_chr_vec provides the object to be updated. The function returns an abbreviation (a tibble).

Usage

```
update_abbr_lup_tb(
  abbr_tb,
  short_name_chr_vec,
  long_name_chr_vec,
  no_plural_chr_vec = NA_character_,
  custom_plural_ls = NULL,
  pfx_rgx_chr = NA_character_
)
```

Arguments

abbr_tb	Abbreviation (a tibble)
short_name_chr_vec	Short name (a character vector)
long_name_chr_vec	Long name (a character vector)
no_plural_chr_vec	No plural (a character vector), Default: 'NA'
custom_plural_ls	Custom plural (a list), Default: NULL
pfx_rgx_chr	Prefix regular expression (a character vector of length 1), Default: 'NA'

Value

Abbreviation (a tibble)

update_first_word_case_chr
Update first word case

Description

update_first_word_case_chr() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update first word case. Function argument phrase_chr specifies the object to be updated. Argument fn provides the object to be updated. The function returns a phrase (a character vector of length 1).

Usage

```
update_first_word_case_chr(phrase_chr, fn = tolower)
```

Arguments

phrase_chr	Phrase (a character vector of length 1)
fn	Function (a function), Default: tolower

Value

Phrase (a character vector of length 1)

update_fns_dmt_tb_chr_vars_chr_vec

Update functions documentation tibble character vector of length 1 vars

Description

update_fns_dmt_tb_chr_vars_chr_vec() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update functions documentation a tibble character vector of length 1 vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_chr provides the object to be updated. The function returns a functions documentation (a tibble).

Usage

```
update_fns_dmt_tb_chr_vars_chr_vec(fns_dmt_tb, data_chr, new_ls, append_lgl)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
data_chr	Data (a character vector of length 1)
new_ls	New (a list)
append_lgl	Append (a logical vector of length 1)

Value

Functions documentation (a tibble)

update_fns_dmt_tb_lgl_vars_chr_vec

Update functions documentation tibble logical vector of length 1 vars

Description

update_fns_dmt_tb_lgl_vars_chr_vec() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update functions documentation a tibble logical vector of length 1 vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_chr provides the object to be updated. The function returns a functions documentation (a tibble).

Usage

```
update_fns_dmt_tb_lgl_vars_chr_vec(fns_dmt_tb, data_chr, new_ls)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
data_chr	Data (a character vector of length 1)
new_ls	New (a list)

Value

Functions documentation (a tibble)

```
update_fns_dmt_tb_ls_vars_chr_vec
```

Update functions documentation tibble list vars

Description

update_fns_dmt_tb_ls_vars_chr_vec() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update functions documentation a tibble list vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_chr provides the object to be updated. The function returns a functions documentation (a tibble).

Usage

```
update_fns_dmt_tb_ls_vars_chr_vec(fns_dmt_tb, data_chr, new_ls, append_lgl)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
data_chr	Data (a character vector of length 1)
new_ls	New (a list)
append_lgl	Append (a logical vector of length 1)

Value

Functions documentation (a tibble)

```
update_fns_dmt_tb_tb    Update functions documentation tibble
```

Description

update_fns_dmt_tb_tb() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update functions documentation a tibble. Function argument fns_dmt_tb specifies the object to be updated. Argument title_ls provides the object to be updated. The function returns a functions documentation (a tibble).

Usage

```
update_fns_dmt_tb_tb(
  fns_dmt_tb,
  title_ls = NULL,
  desc_ls = NULL,
  details_ls = NULL,
  export_ls = NULL,
  output_ls = NULL,
  example_ls = NULL,
  args_ls_ls = NULL,
  append_lgl = T
)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
title_ls	Title (a list), Default: NULL
desc_ls	Description (a list), Default: NULL
details_ls	Details (a list), Default: NULL
export_ls	Export (a list), Default: NULL
output_ls	Output (a list), Default: NULL
example_ls	Example (a list), Default: NULL
args_ls_ls	Arguments (a list of lists), Default: NULL
append_lgl	Append (a logical vector of length 1), Default: T

Value

Functions documentation (a tibble)

update_fn_dmt_chr	<i>Update function documentation</i>
-------------------	--------------------------------------

Description

update_fn_dmt_chr() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update function documentation. Function argument fn_tags_spine_ls specifies the object to be updated. Argument new_tag_chr_ls provides the object to be updated. The function returns a function documentation (a character vector of length 1).

Usage

```
update_fn_dmt_chr(
  fn_tags_spine_ls,
  new_tag_chr_ls,
  fn_name_chr,
  fn_type_chr,
  import_chr_vec
)
```

Arguments

fn_tags_spine_ls Function tags spine (a list)

new_tag_chr_ls New tag (a list of character vectors of length 1)

fn_name_chr Function name (a character vector of length 1)

fn_type_chr Function type (a character vector of length 1)

import_chr_vec Import (a character vector)

Value

Function documentation (a character vector of length 1)

update_fn_dmt_with_slots_chr
Update function documentation with slots

Description

update_fn_dmt_with_slots_chr() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update function documentation with slots. Function argument fn_name_chr specifies the object to be updated. Argument fn_dmt_chr provides the object to be updated. The function returns a function documentation (a character vector of length 1).

Usage

```
update_fn_dmt_with_slots_chr(fn_name_chr, fn_dmt_chr)
```

Arguments

fn_name_chr Function name (a character vector of length 1)

fn_dmt_chr Function documentation (a character vector of length 1)

Value

Function documentation (a character vector of length 1)

update_ns_chr	<i>Update namespace</i>
---------------	-------------------------

Description

update_ns_chr() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to an update namespace. Function argument package_chr specifies the object to be updated. Argument NA provides the object to be updated. The function returns a package name (a character vector of length 1).

Usage

```
update_ns_chr(package_chr)
```

Arguments

package_chr Package (a character vector of length 1)

Value

Package name (a character vector of length 1)

write_all_tbs_in_tbs_r4_to_csvs	<i>Write all tibbles in tibbles readyforwhatsnext S4 to csvs</i>
---------------------------------	--

Description

write_all_tbs_in_tbs_r4_to_csvs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write all a tibbles in a tibbles readyforwhatsnext S4 to csvs. The function is called for its side effects and does not return a value.

Usage

```
write_all_tbs_in_tbs_r4_to_csvs(tbs_r4, r4_name_chr, lup_dir_chr, pfx_chr)
```

Arguments

tbs_r4	Tibbles (a readyforwhatsnext S4)
r4_name_chr	Readyforwhatsnext S4 name (a character vector of length 1)
lup_dir_chr	Lookup table directory (a character vector of length 1)
pfx_chr	Prefix (a character vector of length 1)

write_and_doc_ds_R	<i>Write and document dataset</i>
--------------------	-----------------------------------

Description

write_and_doc_ds_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and a document dataset R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_and_doc_ds_R(
  db,
  overwrite_lgl = T,
  db_chr,
  title_chr,
  desc_chr,
  format_chr = "A tibble",
  url_chr = NA_character_,
  vars_ls = NULL,
  R_dir_chr = "R",
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

Arguments

db	PARAM_DESCRIPTION
overwrite_lgl	Overwrite (a logical vector of length 1), Default: T
db_chr	Database (a character vector of length 1)
title_chr	Title (a character vector of length 1)
desc_chr	Description (a character vector of length 1)
format_chr	Format (a character vector of length 1), Default: 'A tibble'
url_chr	Url (a character vector of length 1), Default: 'NA'
vars_ls	Vars (a list), Default: NULL
R_dir_chr	R directory (a character vector of length 1), Default: 'R'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL
object_type_lup	Object type (a lookup table), Default: NULL

write_and_doc_fn_fls_R

Write and document function files

Description

write_and_doc_fn_fls_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and a document function files R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_and_doc_fn_fls_R(
  fns_dmt_tb,
  r_dir_chr = "R",
  path_to_user_dmt_dir_chr = "../.../Documentation/Code/User",
  path_to_dvpr_dmt_dir_chr = "../.../Documentation/Code/Developer"
)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
r_dir_chr	R directory (a character vector of length 1), Default: 'R'
path_to_user_dmt_dir_chr	Path to user documentation directory (a character vector of length 1), Default: '../.../Documentation/Code/User'
path_to_dvpr_dmt_dir_chr	Path to developer documentation directory (a character vector of length 1), Default: '../.../Documentation/Code/Developer'

write_documented_fns_R

Write documented functions

Description

write_documented_fns_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write documented a functions R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_documented_fns_R(tmp_fn_dir_chr, R_dir_chr)
```

Arguments

tmp_fn_dir_chr	Temporary function directory (a character vector of length 1)
R_dir_chr	R directory (a character vector of length 1)

write_ds_dmt_R	<i>Write dataset documentation</i>
----------------	------------------------------------

Description

write_ds_dmt_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a dataset documentation R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_ds_dmt_R(
  db,
  db_chr,
  title_chr,
  desc_chr,
  format_chr = "A tibble",
  url_chr = NA_character_,
  vars_ls = NULL,
  R_dir_chr = "R",
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

Arguments

db	PARAM_DESCRIPTION
db_chr	Database (a character vector of length 1)
title_chr	Title (a character vector of length 1)
desc_chr	Description (a character vector of length 1)
format_chr	Format (a character vector of length 1), Default: 'A tibble'
url_chr	Url (a character vector of length 1), Default: 'NA'
vars_ls	Vars (a list), Default: NULL
R_dir_chr	R directory (a character vector of length 1), Default: 'R'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL
object_type_lup	Object type (a lookup table), Default: NULL

write_fn_dmt

*Write function documentation***Description**

write_fn_dmt() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a function documentation. The function is called for its side effects and does not return a value.

Usage

```
write_fn_dmt(
  fn_name_chr,
  fn_type_chr,
  fn = NULL,
  fn_desc_chr = NA_character_,
  fn_out_type_chr = NA_character_,
  fn_title_chr = NA_character_,
  example_lgl = F,
  export_lgl = T,
  class_name_chr = "",
  details_chr = "DETAILS",
  args_ls = NULL,
  import_chr_vec = NA_character_,
  doc_in_class_lgl = F,
  object_type_lup = NULL
)
```

Arguments

fn_name_chr	Function name (a character vector of length 1)
fn_type_chr	Function type (a character vector of length 1)
fn	Function (a function), Default: NULL
fn_desc_chr	Function description (a character vector of length 1), Default: 'NA'
fn_out_type_chr	Function out type (a character vector of length 1), Default: 'NA'
fn_title_chr	Function title (a character vector of length 1), Default: 'NA'
example_lgl	Example (a logical vector of length 1), Default: F
export_lgl	Export (a logical vector of length 1), Default: T
class_name_chr	Class name (a character vector of length 1), Default: ''
details_chr	Details (a character vector of length 1), Default: 'DETAILS'
args_ls	Arguments (a list), Default: NULL
import_chr_vec	Import (a character vector), Default: 'NA'
doc_in_class_lgl	Document in class (a logical vector of length 1), Default: F
object_type_lup	Object type (a lookup table), Default: NULL

write_fn_fl_R	<i>Write function file</i>
---------------	----------------------------

Description

write_fn_fl_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a function file R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_fn_fl_R(fns_dmt_tb, r_dir_chr = "R", document_unexp_lgl = T)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
r_dir_chr	R directory (a character vector of length 1), Default: 'R'
document_unexp_lgl	Document unexported (a logical vector of length 1), Default: T

write_from_tmp_R	<i>Write from temporary</i>
------------------	-----------------------------

Description

write_from_tmp_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write from temporary R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_from_tmp_R(
  temp_path_chr,
  dest_path_chr,
  edit_fn = function(x) { x },
  args_ls = NULL
)
```

Arguments

temp_path_chr	Temp path (a character vector of length 1)
dest_path_chr	Dest path (a character vector of length 1)
edit_fn	Edit (a function), Default: function(x) x
args_ls	Arguments (a list), Default: NULL

write_nsimps_to_desc	<i>Write namespace imports to description</i>
----------------------	---

Description

write_nsimps_to_desc() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a namespace imports to a description. The function is called for its side effects and does not return a value.

Usage

```
write_nsimps_to_desc(dev_pkgs_chr_vec = NA_character_)
```

Arguments

dev_pkgs_chr_vec
Dev packages (a character vector), Default: 'NA'

write_pkg_R	<i>Write package</i>
-------------	----------------------

Description

write_pkg_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a package R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_pkg_R(package_chr, R_dir_chr = "R")
```

Arguments

package_chr Package (a character vector of length 1)
R_dir_chr R directory (a character vector of length 1), Default: 'R'

write_pt_lup_db_R	<i>Write prototype lookup table database</i>
-------------------	--

Description

write_pt_lup_db_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a prototype lookup table database R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_pt_lup_db_R(R_dir_chr = "R")
```

Arguments

R_dir_chr	R directory (a character vector of length 1), Default: 'R'
-----------	--

write_std_imp_R	<i>Write standard import</i>
-----------------	------------------------------

Description

write_std_imp_R() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a standard import R. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_std_imp_R(R_dir_chr = "R")
```

Arguments

R_dir_chr	R directory (a character vector of length 1), Default: 'R'
-----------	--

write_tb_to_csv	<i>Write tibble to csv</i>
-----------------	----------------------------

Description

write_tb_to_csv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a tibble to csv. The function is called for its side effects and does not return a value.

Usage

```
write_tb_to_csv(tbs_r4, slot_nm_chr, r4_name_chr, lup_dir_chr, pfx_chr)
```

Arguments

tbs_r4	Tibbles (a readyforwhatsnext S4)
slot_nm_chr	Slot name (a character vector of length 1)
r4_name_chr	Readyforwhatsnext S4 name (a character vector of length 1)
lup_dir_chr	Lookup table directory (a character vector of length 1)
pfx_chr	Prefix (a character vector of length 1)

write_ws	<i>Write workspace</i>
----------	------------------------

Description

write_ws() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write a workspace. The function is called for its side effects and does not return a value.

Usage

```
write_ws(path_chr)
```

Arguments

path_chr	Path (a character vector of length 1)
----------	---------------------------------------

%>%	<i>Pipe Implements: https://github.com/sckott/analogsea/issues/32 and https://github.com/rstudio/ggvis/blob/master/R/pipe.R.</i>
-----	--

Description

Pipe Implements: <https://github.com/sckott/analogsea/issues/32> and <https://github.com/rstudio/ggvis/blob/master/R/pipe.R>

Arguments

lhs, rhs	A visualisation and a function to apply to it
----------	---

Index

* datasets

abbreviations_lup, [4](#)
fn_type_lup_tb, [7](#)
object_type_lup, [25](#)
%>%, [43](#)

abbreviations_lup, [4](#)
add_indef_artl_to_item_chr_vec, [5](#)
add_indefartls_to_phrases_chr_vec, [4](#)
add_plurals_to_abbr_lup_tb, [6](#)
assert_does_not_match_terms, [6](#)

close_open_sinks, [7](#)

fn_type_lup_tb, [7](#)
force_req_pkg_install, [8](#)

get_fn_args_chr_vec, [8](#)
get_fn_nms_in_file_chr, [9](#)
get_from_lup_obj, [9](#)
get_outp_obj_type_chr_vec, [10](#)
get_r4_obj_slots_chr_vec, [10](#)
get_return_obj_nm_chr, [11](#)

import_xls_sheets_ls, [11](#)

make_abbr_lup_tb, [12](#)
make_and_doc_fn_type_R, [13](#)
make_arg_desc_chr_vec, [13](#)
make_arg_desc_ls, [14](#)
make_arg_desc_spine_chr, [14](#)
make_arg_title_chr_vec, [15](#)
make_arg_type_abbr_chr_vec, [16](#)
make_arg_type_abbr_spine_chr, [16](#)
make_arg_type_chr_vec, [17](#)
make_arg_type_lup_ls, [17](#)
make_fn_desc_chr_vec, [18](#)
make_fn_desc_spine_chr_vec, [18](#)
make_fn_dmt_spine_chr_ls, [19](#)
make_fn_dmt_tbl_tb, [20](#)
make_fn_dmt_tbl_tpl_tb, [20](#)
make_fn_title_chr_vec, [21](#)
make_fn_type_lup_tb, [21](#)
make_gtr_str_dmt_spine_chr_ls, [22](#)
make_new_fn_dmt_chr_ls, [22](#)

make_obj_lup_tb, [23](#)
make_std_fn_dmt_spine_chr_ls, [24](#)

object_type_lup, [25](#)

read_fns, [25](#)
ready4fun (ready4fun-package), [3](#)
ready4fun-package, [3](#)
remove_collate_chr_vec, [26](#)
remove_obj_type_from_nm_chr_vec, [26](#)
replace_abbr_chr, [27](#)
reset_pkg_files_R, [27](#)
rowbind_all_tbs_in_r4_obj_r4, [28](#)
rowbind_tbs_in_r4_obj_r4, [28](#)

unload_packages, [29](#)
update_abbr_lup_tb, [29](#)
update_first_word_case_chr, [30](#)
update_fn_dmt_chr, [33](#)
update_fn_dmt_with_slots_chr, [34](#)
update_fns_dmt_tb_chr_vars_chr_vec, [31](#)
update_fns_dmt_tb_lgl_vars_chr_vec, [31](#)
update_fns_dmt_tb_ls_vars_chr_vec, [32](#)
update_fns_dmt_tb_tb, [32](#)
update_ns_chr, [35](#)

write_all_tbs_in_tbs_r4_to_csvs, [35](#)
write_and_doc_ds_R, [36](#)
write_and_doc_fn_fls_R, [37](#)
write_documented_fns_R, [37](#)
write_ds_dmt_R, [38](#)
write_fn_dmt, [39](#)
write_fn_fl_R, [40](#)
write_from_tmp_R, [40](#)
write_nsimps_to_desc, [41](#)
write_pkg_R, [41](#)
write_pt_lup_db_R, [42](#)
write_std_imp_R, [42](#)
write_tb_to_csv, [43](#)
write_ws, [43](#)