Package: kwb.test (via r-universe)

September 15, 2024

Type Package

Title Test whether Different Functions Return the Same

Version 0.1.0.9000

Description Functions to test whether different functions return the same results. I use these functions to test whether a function still returns the same result after cleaning code.

License MIT + file LICENSE

URL https://github.com/kwb-r/kwb.test

BugReports https://github.com/kwb-r/kwb.test/issues

Imports kwb.utils, usethis

Suggests compare, testthat

Remotes github::kwb-r/kwb.utils

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

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

RemoteUrl https://github.com/KWB-R/kwb.test

RemoteRef HEAD

RemoteSha ddfb2dcf9de81ae1f8f0afad1404f9f032b8a69a

Contents

create_test_files *Create Test Files*

Description

Create test files for each source file containing one `test_that` call for each function in the package

Usage

```
create_test_files(package_dir = getwd(), target_dir = NULL,
  file_per_function = TRUE, full = FALSE, dbg = TRUE)
```

Arguments

package_dir	path to package directory in which to create the test files
target_dir	directory in which to create the test files. Defaults to <package_dir>/tests/testthat.
file_per_function	if TRUE (default), one test file test-<function>.R is generated for each function, otherwise one test file test-<source-file> is generated for each source file.
full	if TRUE, test calls with many argument combinations are generated instead of only one call
dbg	if TRUE, debug messages are shown

Details

Existing test files will not be overwritten.

loadArgs *Load Function Arguments and Results From Files*

Description

Read the function arguments and function results that were stored in RData files (in objects `args` and `result`, respectively) by a previous call of `saveArgs`.

Usage

```
loadArgs(functionName = NULL, data.dir = file.path(tempdir(), "test"))
```

Arguments

functionName	Name of the function to load arguments and results for. The name is used to create a search pattern for RData files in data.dir.
data.dir	Directory in which to look for RData files matching args_<functionName>_<hhmmss>_<no>.RData. The default is the subfolder test in tempdir().

Value

list with as many items as there were files args_<functionName>_* in the directory given in `data.dir`. Each list element has two components: `args` containing the arguments that were given to the function and `result` containing what the function returned.

See Also

saveArgs

Examples

```
# Define a function that stores its arguments and result with saveArgs
double <- function(x) {
  result <- 2 * x
  saveArgs("double", args = list(x = x), result = result)
  result
}

# Set global variable TESTMODE to "activate" saveArgs() in double()
TESTMODE <- TRUE

# Call the function a couple of times
double(4)
double(-99)
double(1:10)

# Load what was stored behind the scenes
testdata <- loadArgs("double")

# "Deactivate" saveArgs() in double()
TESTMODE <- FALSE

# Rerun the function with the stored arguments
results <- lapply(testdata, function(x) do.call("double", x$args))

# Compare the new with the old results
identical(results, lapply(testdata, "[[", "result"))
```

---

printTestMessage *Print a Test with its Result*

Description

Print a test with its result as a message and return the message as a character string

Usage

```
printTestMessage(testexpression, testresult, newline = TRUE)
```

Arguments

testexpression	text description of what was tested
testresult	boolean result (of length one) of the test
newline	if TRUE (default) a new line character is appended to the message shown.

Value

the message that was shown as a character string

Examples

```
printTestMessage("apple == apple", 1 == 1)
printTestMessage("apple == pear", 1 == 2)
```

---

saveArgs *Save the Arguments and Result of a Function Call*

Description

Save the list of named arguments given in ... to an RData file args_<functionName>_<hhmmss>_<no>.RData in the directory given in targetdir. This function can be used to log the inputs given to a function together with the result returned by the function. test_function can then be used to check whether another version of the function (e.g. obtained by code cleaning) can reproduce the stored results from the stored arguments. Check out the example on the help page for test_function.

Usage

```
saveArgs(functionName, ...,
  targetdir = kwb.utils::createDirectory(file.path(tempdir(), "test")))
```

Arguments

functionName	name of the function to which the arguments to be saved belong. It will be used to generate a file name for the RData file.
...	named arguments representing the arguments that have been given to the function functionName.
targetdir	directory in which to store the objects given in ... Default: subdirectory test in tempdir()

Value

path to the file written (invisibly)

See Also

loadArgs

testColumnwiseIdentity *Check Corresponding Columns in two Data Frames for Identity*

Description

For all columns in the first data frame, check if the second data frame has identical values in columns of the same name

Usage

```
testColumnwiseIdentity(...)
```

Arguments

...	two data frames given as named arguments. The argument names will appear in the output. By doing so you can give a longer expression that returns a data frame a short name 'on-the-fly'.

Examples

```
# Compare two identical data frames. Give them short names data.1 and data.2
testColumnwiseIdentity(data.1 = (x <- data.frame(a = 1:2, b = 2:3)),
                       data.2 = x)

# Compare two data frames differing in one column
testColumnwiseIdentity(A = data.frame(x = 1:2, y = 2:3),
                       B = data.frame(x = 1:2, y = 3:4))
```

test_function *Test if Function Reproduces Stored Results*

Description

Call the function `functionName` with the arguments contained in `testdata` and compare the results with the results in `testdata` for identity.

Usage

```
test_function(functionName, testdata = loadArgs(functionName,
  file.path(tempdir(), "test")), dbg = TRUE)
```

Arguments

functionName	Name of the function to test. It must be callable, i.e. either defined in the global environment or on the search path.
testdata	List of lists containing function arguments (in elemenet args) and results (in element result), just as returned by loadArgs. If no testdata are given, it is tried to load test data by calling loadArgs on functionName.
dbg	if TRUE (default) debug messages are shown

Value

TRUE If the function functionName is able to reproduce the same results as given in the result elements in testdata for all the argument combinations given in the args elements in testdata.

Examples

```
# Define a function using saveArgs() to save arguments and result
squareSum <- function(a, b) {
  result <- a * a + b * b
  saveArgs("squareSum", args = list(a = a, b = b), result = result)
  result
}

# Set global variable TESTMODE to "activate" saveArgs() in squareSum()
TESTMODE <- TRUE

# Call the function with different arguments
squareSum(1, 2)
squareSum(2, 3)
squareSum(-1, -2)

# The arguments and function results were saved here:
dir(file.path(tempdir(), "test"))

# Write a new (wrong) version of the function
squareSum.new <- function(a, b) {
  a * a - b * b
}

# Check if it returns the same results
test_function("squareSum.new", loadArgs("squareSum"))

# If no test data are given, loadArgs is called on the function to test,
# i.e. testing squareSum on the test data created by the same function will
# return TRUE if the function did not change in the meanwhile.
test_function("squareSum")
```

Index