

Package: autotest (via r-universe)

October 23, 2024

Title Automatic Package Testing

Version 0.0.2.211

Description Automatic testing of R packages via a simple YAML schema.

License GPL-3

URL <https://docs.ropensci.org/autotest/>,
<https://github.com/ropensci-review-tools/autotest>

BugReports <https://github.com/ropensci-review-tools/autotest/issues>

Imports cli, data.table, digest, here, magrittr, memoise, methods,
pkgload, rlang, testthat, tibble, withr, yaml

Suggests devtools, dplyr, DT, fs, knitr, rmarkdown, roxygen2, tidy,
usethis

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

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

RemoteUrl <https://github.com/ropensci-review-tools/autotest>

RemoteRef main

RemoteSha fc80d9f8028bf288515976fc9c37477befb90b79

Contents

at_yaml_template	2
autotest_obj	2
autotest_package	3
autotest_types	5
autotest_yaml	6

examples_to_yaml	7
expect_autotest_notes	8
expect_autotest_no_err	8
expect_autotest_no_testdata	9
expect_autotest_no_warn	9
expect_autotest_testdata	10

Index	11
--------------	-----------

at_yaml_template	<i>at_yaml_template</i>
------------------	-------------------------

Description

Generate a 'yaml' template for an 'autotest'.

Usage

```
at_yaml_template(loc = tempdir())
```

Arguments

loc	Location to generate template file. Append with filename and '.yaml' suffix to overwrite default name of 'autotest.yaml', otherwise this parameter will be used to specify directory only.
-----	--

See Also

Other yaml: [autotest_yaml\(\)](#), [examples_to_yaml\(\)](#)

autotest_obj	<i>autotest_obj class definition</i>
--------------	--------------------------------------

Description

This function exists only to provide the class definitions for test objects, and is not intended to be called directly.

Usage

```

autotest_obj(
  package = NA_character_,
  package_loc = NULL,
  test_name = NA_character_,
  fn_name = NA_character_,
  parameters = list(),
  parameter_types = NA_character_,
  class = NULL,
  classes = NULL,
  env = new.env(),
  test = FALSE,
  quiet = FALSE
)

```

Arguments

package	Name of package for which object is to be constructed.
package_loc	Location of package on local system (for source packages only)
test_name	Name of test (use autotest_types to get all test names).
fn_name	Name of function to be tested.
parameters	Names of all parameters for that function.
parameter_types	Types of input parameters.
class	Class of an individual parameter.
classes	Classes of all parameters.
env	Environment in which tests are to be run.
test	If FALSE, return only descriptions of tests which would be run with test = TRUE, without actually running them.
quiet	If FALSE, issue progress and other messages during testing of object.

autotest_package	<i>autotest_package</i>
------------------	-------------------------

Description

Automatically test an entire package by converting examples to yaml format and submitting each to the [autotest_yaml](#) function.

Usage

```

autotest_package(
  package = ".",
  functions = NULL,
  exclude = NULL,
  test = FALSE,
  test_data = NULL,
  quiet = FALSE
)

```

Arguments

package	Name of package, as either <ol style="list-style-type: none"> 1. Path to local package source 2. Name of installed package 3. Full path to location of installed package if not on <code>.libPaths</code>, or 4. Default which presumes current directory is within package to be tested.
functions	Optional character vector containing names of functions of nominated package to be included in 'autotesting'.
exclude	Optional character vector containing names of any functions of nominated package to be excluded from 'autotesting'.
test	If FALSE, return only descriptions of tests which would be run with test = TRUE, without actually running them.
test_data	Result returned from calling either <code>autotest_types</code> or <code>autotest_package</code> with test = FALSE that contains a list of all tests which would be conducted. These tests have an additional flag, test, which defaults to TRUE. Setting any tests to FALSE will avoid running them when test = TRUE.
quiet	If 'FALSE', provide printed output on screen.

Value

An `autotest_package` object which is derived from a **tibble** `tbl_df` object. This has one row for each test, and the following nine columns:

1. `type` The type of result, either "dummy" for test = FALSE, or one of "error", "warning", "diagnostic", or "message".
2. `test_name` Name of each test
3. `fn_name` Name of function being tested
4. `parameter` Name of parameter being tested
5. `parameter_type` Expected type of parameter as identified by autotest.
6. `operation` Description of the test
7. `content` For test = FALSE, the expected behaviour of the test; for test = TRUE, the observed discrepancy with that expected behaviour
8. `test` If FALSE (default), list all tests without implementing them, otherwise implement all tests.

9. `yaml_hash` A unique hash which may be used to extract the yml specification of each test.

Some columns may contain NA values, as explained in the Note.

Note

Some columns may contain NA values, including:

- `parameter` and `parameter_type`, for tests applied to entire functions, such as tests of return values.
- `test_name` for warnings or errors generated through "normal" function calls generated directly from example code, in which case type will be "warning" or "error", and content will contain the content of the corresponding message.

See Also

Other main_functions: [autotest_types\(\)](#)

<code>autotest_types</code>	<i>autotest_types</i>
-----------------------------	-----------------------

Description

List all types of 'autotests' currently implemented.

Usage

```
autotest_types(notest = NULL)
```

Arguments

<code>notest</code>	Character string of names of tests which should be switched off by setting the test column to FALSE. Run this function first without this parameter to get all names, then re-run with this parameter switch specified tests off.
---------------------	---

Value

An autotest object with each row listing one unique type of test which can be applied to every parameter (of the appropriate class) of each function.

See Also

Other main_functions: [autotest_package\(\)](#)

autotest_yaml	<i>autotest_yaml</i>
---------------	----------------------

Description

Automatically test inputs to functions specified in a 'yaml' template.

Usage

```
autotest_yaml(  
  yaml = NULL,  
  filename = NULL,  
  test = TRUE,  
  test_data = NULL,  
  quiet = FALSE  
)
```

Arguments

yaml	A 'yaml' template as a character vector, either hand-coded or potentially loaded via readLines function or similar. Should generally be left at default of 'NULL', with template specified by 'filename' parameter.
filename	Name (potentially including path) of file containing 'yaml' template. See at_yaml_template for details of template. Default uses template generated by that function, and held in local './tests' directory.
test	If FALSE, return only descriptions of tests which would be run with test = TRUE, without actually running them.
test_data	Result returned from calling either autotest_types or autotest_package with test = FALSE that contains a list of all tests which would be conducted. These tests have an additional flag, test, which defaults to TRUE. Setting any tests to FALSE will avoid running them when test = TRUE.
quiet	If 'FALSE', provide printed output on screen.

Value

An autotest_pkg object, derived from a **tibble**, detailing instances of unexpected behaviour for every parameter of every function.

See Also

Other yaml: [at_yaml_template\(\)](#), [examples_to_yaml\(\)](#)

Examples

```
## Not run:
yaml_list <- examples_to_yaml (package = "stats", functions = "reshape")
res <- autotest_yaml (yaml = yaml_list)

## End(Not run)
```

examples_to_yaml	<i>examples_to_yaml</i>
------------------	-------------------------

Description

Convert examples for a specified package, optionally restricted to one or more specified functions, to a list of 'autotest' 'yaml' objects to use to automatically test package.

Usage

```
examples_to_yaml(  
  package = NULL,  
  functions = NULL,  
  exclude = NULL,  
  quiet = FALSE  
)
```

Arguments

package	Name of package, as either <ol style="list-style-type: none">1. Path to local package source2. Name of installed package3. Full path to location of installed package if not on .libPaths, or4. Default which presumes current directory is within package to be tested.
functions	If specified, names of functions from which examples are to be obtained.
exclude	Names of functions to exclude from 'yaml' template
quiet	If 'FALSE', provide printed output on screen.

See Also

Other yaml: [at_yaml_template\(\)](#), [autotest_yaml\(\)](#)

expect_autotest_notes *expect_autotest_notes*

Description

Expect test_data param of autotest_package to have additional note column explaining why tests have been switched off.

Usage

expect_autotest_notes(object)

Arguments

object An autotest object to be tested

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_testdata\(\)](#)

expect_autotest_no_err
expect_autotest_no_err

Description

Expect autotest_package() to be clear of errors

Usage

expect_autotest_no_err(object)

Arguments

object An autotest object to be tested

Value

(invisibly) The same object

See Also

Other expectations: [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

`expect_autotest_no_testdata`
expect_autotest_no_testdata

Description

Expect `autotest_package()` to be clear of errors with no tests switched off

Usage

`expect_autotest_no_testdata(object = NULL)`

Arguments

`object` Not used here, but required for test that expectations

Value

(invisibly) The autotest object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

`expect_autotest_no_warn`
expect_autotest_no_warn

Description

Expect `autotest_package()` to be clear of warnings

Usage

`expect_autotest_no_warn(object)`

Arguments

`object` An autotest object to be tested

Value

(invisibly) The same object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_notes\(\)](#), [expect_autotest_testdata\(\)](#)

expect_autotest_testdata

expect_autotest_testdata

Description

Expect `autotest_package()` to be clear of errors with some tests switched off, and to have note column explaining why those tests are not run.

Usage

```
expect_autotest_testdata(object)
```

Arguments

`object` An `autotest_package` object with a test column flagging tests which are not to be run on the local package.

Value

(invisibly) The autotest object

See Also

Other expectations: [expect_autotest_no_err\(\)](#), [expect_autotest_no_testdata\(\)](#), [expect_autotest_no_warn\(\)](#), [expect_autotest_notes\(\)](#)

Index

- * **class**
 - autotest_obj, [2](#)
- * **expectations**
 - expect_autotest_no_err, [8](#)
 - expect_autotest_no_testdata, [9](#)
 - expect_autotest_no_warn, [9](#)
 - expect_autotest_notes, [8](#)
 - expect_autotest_testdata, [10](#)
- * **main_functions**
 - autotest_package, [3](#)
 - autotest_types, [5](#)
- * **yaml**
 - at_yaml_template, [2](#)
 - autotest_yaml, [6](#)
 - examples_to_yaml, [7](#)
 - .libPaths, [4, 7](#)

[at_yaml_template, 2, 6, 7](#)
[autotest_obj, 2](#)
[autotest_package, 3, 4-6](#)
[autotest_types, 3-5, 5, 6](#)
[autotest_yaml, 2, 3, 6, 7](#)

[examples_to_yaml, 2, 6, 7](#)
[expect_autotest_no_err, 8, 8, 9, 10](#)
[expect_autotest_no_testdata, 8, 9, 10](#)
[expect_autotest_no_warn, 8, 9, 9, 10](#)
[expect_autotest_notes, 8, 8, 9, 10](#)
[expect_autotest_testdata, 8-10, 10](#)

[readLines, 6](#)