Package ‘DBItest’

December 16, 2019

Title Testing 'DBI' 'Backends'

Version 1.7.0

Date 2019-12-15

Description A helper that tests 'DBI' back ends for conformity
to the interface.

License LGPL (>= 2.1)

URL https://dbitest.r-dbi.org, https://github.com/r-dbi/DBItest

BugReports https://github.com/r-dbi/DBItest/issues

Depends R (>= 3.2.0)

Imports blob (>= 1.2.0), callr, DBI (>= 1.1.0), desc, hms (>= 0.5.0),
lubridate, methods, R6, rlang (>= 0.2.0), testthat (>= 2.0.0),
withr

Suggests debugme, devtools, knitr, lintr, rmarkdown, RSQLite

VignetteBuilder knitr

Encoding UTF-8

KeepSource true

LazyData true

RoxygenNote 7.0.2

Collate 'DBItest.R' 'context.R' 'expectations.R' 'import-dbi.R'
'import-testthat.R' 'run.R' 's4.R' 'spec-getting-started.R'
'spec-compliance-methods.R' 'spec-driver-constructor.R'
'spec-driver-data-type.R' 'spec-connection-data-type.R'
'spec-result-create-table-with-data-type.R'
'spec-driver-connect.R' 'spec-connection-disconnect.R'
'spec-result-send-query.R' 'spec-result-fetch.R'
'spec-result-roundtrip.R' 'spec-result-clear-result.R'
'spec-result-get-query.R' 'spec-result-send-statement.R'
'spec-result-execute.R' 'spec-sql-quote-string.R'
'spec-sql-quote-literal.R' 'spec-sql-quote-identifier.R'
'spec-sql-unquote-identifier.R' 'spec-sql-read-table.R'
'spec-sql-create-table.R' 'spec-sql-append-table.R'

'spec-sql-write-table.R' 'spec-sql-list-tables.R'
'spec-sql-exists-table.R' 'spec-sql-remove-table.R'
'spec-sql-list-objects.R' 'spec-meta-bind-runner.R'
'spec-meta-bind-tester-extra.R' 'spec-meta-bind.R'
'spec-meta-bind-.R' 'spec-meta-is-valid.R'
'spec-meta-has-completed.R' 'spec-meta-get-statement.R'
'spec-meta-get-row-count.R' 'spec-meta-get-rows-affected.R'
'spec-transaction-begin-commit-rollback.R'
'spec-transaction-with-transaction.R' 'spec-driver-get-info.R'
'spec-connection-get-info.R' 'spec-sql-list-fields.R'
'spec-meta-column-info.R' 'spec-meta-get-info-result.R'
'spec-driver.R' 'spec-connection.R' 'spec-result.R'
'spec-sql.R' 'spec-meta.R' 'spec-transaction.R'
'spec-compliance.R' 'spec-stress-connection.R' 'spec-stress.R'
'spec-all.R' 'spec-.R' 'test-all.R' 'test-getting-started.R'
'test-driver.R' 'test-connection.R' 'test-result.R'
'test-sql.R' 'test-meta.R' 'test-transaction.R'
'test-compliance.R' 'test-stress.R' 'tweaks.R' 'utf8.R'
'utils.R' 'zzz.R'

NeedsCompilation no

Author Kirill Müller [aut, cre] (<https://orcid.org/0000-0002-1416-3412>),
RStudio [cph],
R Consortium [fnd]

Maintainer Kirill Müller <krlmlr+r@mailbox.org>

Repository CRAN

Date/Publication 2019-12-16 22:20:02 UTC

R topics documented:

`DBItest-package` *DBItest: Testing 'DBI' 'Backends'*

Description

A helper that tests 'DBI' back ends for conformity to the interface.

Details

The two most important functions are `make_context()` and `test_all()`. The former tells the package how to connect to your DBI backend, the latter executes all tests of the test suite. More fine-grained test functions (all with prefix `test_`) are available.

See the package's vignette for more details.

Author(s)

Kirill Müller

See Also

Useful links:

- `https://dbitest.r-dbi.org`
- `https://github.com/r-dbi/DBItest`
- Report bugs at `https://github.com/r-dbi/DBItest/issues`

`make_context` *Test contexts*

Description

Create a test context, set and query the default context.

Usage

```
make_context(
  drv,
  connect_args = NULL,
  set_as_default = TRUE,
  tweaks = NULL,
  name = NULL,
  default_skip = NULL
)

set_default_context(ctx)

get_default_context()
```

Arguments

drv	[DBIConnector] An object of class DBIConnector that describes how to connect to the database.
connect_args	[named list] Deprecated.
set_as_default	[logical(1)] Should the created context be set as default context?
tweaks	[DBItest_tweaks] Tweaks as constructed by the `tweaks()` function.
name	[character] An optional name of the context which will be used in test messages.
default_skip	[character] Default value of `skip` argument to `test_all()` and other testing functions.
ctx	[DBItest_context] A test context.

Value

[DBItest_context]
A test context, for `set_default_context` the previous default context (invisibly) or `NULL`.

Examples

```
make_context(
  new(
    "DBIConnector",
    .drv = RSQLite::SQLite(),
    .conn_args = list(dbname = tempfile("DBItest", fileext = ".sqlite"))
  ),
  tweaks = DBItest::tweaks(
    constructor_relax_args = TRUE,
    placeholder_pattern = c("?", "$1", "$name", ":name"),
    date_cast = function(x) paste0("'", x, "'"),
    time_cast = function(x) paste0("'", x, "'"),
    timestamp_cast = function(x) paste0("'", x, "'"),
    logical_return = function(x) as.integer(x),
    date_typed = FALSE,
    time_typed = FALSE,
    timestamp_typed = FALSE
  ),
  default_skip = c("roundtrip_date", "roundtrip_timestamp")
)
```

`test_all` *Run all tests*

Description

`test_all()` calls all tests defined in this package (see the section "Tests" below). This function supports running only one test by setting an environment variable, e.g., set the `DBITEST_ONLY_RESULT` to a nonempty value to run only `test_result()`.

`test_some()` allows testing one or more tests.

Usage

```
test_all(skip = NULL, run_only = NULL, ctx = get_default_context())

test_some(test, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.
`test`	[character] A character vector of regular expressions describing the tests to run. The regular expressions are matched against the entire test name.

Details

Internally `^` and `$` are used as prefix and suffix around the regular expressions passed in the `skip` and `run_only` arguments.

Tests

This function runs the following tests, except the stress tests:

`test_getting_started()`: Getting started with testing

`test_driver()`: Test the "Driver" class

`test_connection()`: Test the "Connection" class

`test_result()`: Test the "Result" class

`test_sql()`: Test SQL methods

`test_meta()`: Test metadata functions

`test_transaction()`: Test transaction functions

`test_compliance()`: Test full compliance to DBI

`test_stress()`: Stress tests (not tested with `test_all`)

`test_compliance` *Test full compliance to DBI*

Description

Test full compliance to DBI

Usage

```
test_compliance(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_connection()`, `test_driver()`, `test_getting_started()`, `test_meta()`, `test_result()`, `test_sql()`, `test_stress()`, `test_transaction()`

`test_connection` *Test the "Connection" class*

Description

Test the "Connection" class

Usage

```
test_connection(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_driver()`, `test_getting_started()`, `test_meta()`, `test_result()`, `test_sql()`, `test_stress()`, `test_transaction()`

`test_driver` *Test the "Driver" class*

Description

Test the "Driver" class

Usage

```
test_driver(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_connection()`, `test_getting_started()`, `test_meta()`, `test_result()`, `test_sql()`, `test_stress()`, `test_transaction()`

`test_getting_started` *Getting started with testing*

Description

Tests very basic features of a DBI driver package, to support testing and test-first development right from the start.

Usage

```
test_getting_started(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_connection()`, `test_driver()`, `test_meta()`, `test_result()`, `test_sql()`, `test_stress()`, `test_transaction()`

`test_meta` *Test metadata functions*

Description

Test metadata functions

Usage

```
test_meta(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_connection()`, `test_driver()`, `test_getting_started()`, `test_result()`, `test_sql()`, `test_stress()`, `test_transaction()`

`test_result` *Test the "Result" class*

Description

Test the "Result" class

Usage

```
test_result(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance`(), `test_connection`(), `test_driver`(), `test_getting_started`(), `test_meta`(), `test_sql`(), `test_stress`(), `test_transaction`()

`test_sql` *Test SQL methods*

Description

Test SQL methods

Usage

```
test_sql(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_connection()`, `test_driver()`, `test_getting_started()`, `test_meta()`, `test_result()`, `test_stress()`, `test_transaction()`

`test_transaction` *Test transaction functions*

Description

Test transaction functions

Usage

```
test_transaction(skip = NULL, run_only = NULL, ctx = get_default_context())
```

Arguments

`skip`	[character()] A vector of regular expressions to match against test names; skip test if matching any. The regular expressions are matched against the entire test name.
`run_only`	[character()] A vector of regular expressions to match against test names; run only these tests. The regular expressions are matched against the entire test name.
`ctx`	[DBItest_context] A test context as created by `make_context()`.

See Also

Other tests: `test_compliance()`, `test_connection()`, `test_driver()`, `test_getting_started()`, `test_meta()`, `test_result()`, `test_sql()`, `test_stress()`

`tweaks` *Tweaks for DBI tests*

Description

The tweaks are a way to control the behavior of certain tests. Currently, you need to search the **DBItest** source code to understand which tests are affected by which tweaks. This function is usually called to set the `tweaks` argument in a `make_context()` call.

Usage

```
tweaks(
  ...,
  constructor_name = NULL,
  constructor_relax_args = FALSE,
  strict_identifier = FALSE,
  omit_blob_tests = FALSE,
  current_needs_parens = FALSE,
  union = function(x) paste(x, collapse = " UNION "),
  placeholder_pattern = NULL,
  logical_return = identity,
  date_cast = function(x) paste0("date('", x, "')"),
  time_cast = function(x) paste0("time('", x, "')"),
  timestamp_cast = function(x) paste0("timestamp('", x, "')"),
  blob_cast = identity,
  date_typed = TRUE,
  time_typed = TRUE,
  timestamp_typed = TRUE,
  temporary_tables = TRUE,
  list_temporary_tables = TRUE,
  is_null_check = function(x) paste0("(", x, " IS NULL)")
)
```

Arguments

`...`	[any] Unknown tweaks are accepted, with a warning. The ellipsis also makes sure that you only can pass named arguments.
`constructor_name`	[character(1)] Name of the function that constructs the `Driver` object.
`constructor_relax_args`	[logical(1)] If `TRUE`, allow a driver constructor with default values for all arguments; otherwise, require a constructor with empty argument list (default).
`strict_identifier`	[logical(1)] Set to `TRUE` if the DBMS does not support arbitrarily-named identifiers even when quoting is used.
`omit_blob_tests`	[logical(1)] Set to `TRUE` if the DBMS does not support a `BLOB` data type.
`current_needs_parens`	[logical(1)] Set to `TRUE` if the SQL functions `current_date`, `current_time`, and `current_timestamp` require parentheses.
`union`	[function(character)] Function that combines several subqueries into one so that the resulting query

returns the concatenated results of the subqueries

`placeholder_pattern`	[character] A pattern for placeholders used in `dbBind()`, e.g., `"?"`, `"$1"`, or `":name"`. See `make_placeholder_fun()` for details.
`logical_return`	[function(logical)] A vectorized function that converts logical values to the data type returned by the DBI backend.
`date_cast`	[function(character)] A vectorized function that creates an SQL expression for coercing a string to a date value.
`time_cast`	[function(character)] A vectorized function that creates an SQL expression for coercing a string to a time value.
`timestamp_cast`	[function(character)] A vectorized function that creates an SQL expression for coercing a string to a timestamp value.
`blob_cast`	[function(character)] A vectorized function that creates an SQL expression for coercing a string to a blob value.
`date_typed`	[logical(1L)] Set to `FALSE` if the DBMS doesn't support a dedicated type for dates.
`time_typed`	[logical(1L)] Set to `FALSE` if the DBMS doesn't support a dedicated type for times.
`timestamp_typed`	[logical(1L)] Set to `FALSE` if the DBMS doesn't support a dedicated type for timestamps.
`temporary_tables`	[logical(1L)] Set to `FALSE` if the DBMS doesn't support temporary tables.
`list_temporary_tables`	[logical(1L)] Set to `FALSE` if the DBMS doesn't support listing temporary tables.
`is_null_check`	[function(character)] A vectorized function that creates an SQL expression for checking if a value is `NULL`.

Examples

```
## Not run:
make_context(..., tweaks = tweaks(strict_identifier = TRUE))

## End(Not run)
```

Index