Package: doltr (via r-universe)

October 16, 2024

Title A client for the dolt database

Version 0.0.0.9000

Description Creates a DBI-compliant interface to dolt databases (<https://www.dolthub.com>). Also manages local dolt server processes, provides convenience functions for dolt versioning, and an RStudio connection pane interface.

License AGPL (>= 3)

URL https://ecohealthalliance.github.io/doltr, https://github.com/ecohealthalliance/doltr

BugReports https://github.com/ecohealthalliance/doltr/issues

Imports blob, cli, DBI, dbplyr, dbx, dplyr, httpuv, jsonlite, methods, processx, ps, rlang, RMariaDB, R.utils, rstudioapi, utils

Suggests covr, withr, DBItest, knitr, lintr, rmarkdown, spelling, testthat

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

SystemRequirements dolt

Collate 'cli.R' 'dolt-config.R' 'dolt-connection.R' 'dolt-diffs.R' 'dolt-info.R' 'server.R' 'dolt-local-connection.R' 'dolt-methods.R' 'dolt-nav.R' 'dolt-remote.R' 'dolt-stage-commit.R' 'dolt-tables.R' 'dolt-types.R' 'dolt-vars.R' 'dolt.R' 'doltr-package.R' 'install.R' 'onload.R' 'pane.R' 'query.R' 'read-table.R' 'state.R' 'utils.R' 'write-table.R'

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

RemoteUrl https://github.com/ecohealthalliance/doltr

RemoteRef HEAD

RemoteSha 0ea27d4d73617b167ab02e3220c57e371ecdeb10

Contents

`dbDataType,DoltConnection-method`
Dolt Data Types

Description

`dbDataType` matches R data types to dolt data types. For text and blob data types, it automatically chooses amongst options (e.g., `VARCHAR(N)`, `TEXT`, `LONGTEXT`, `etc.`) based on maximum field length. An attribute of maximum size of these fields is returned to support operations where fields need to be recast in revision.

Usage

```
## S4 method for signature 'DoltConnection'
dbDataType(
  dbObj,
  obj,
  min_varchar = Sys.getenv("DOLT_MINVARCAHR", 255L),
  max_varchar = Sys.getenv("DOLT_MAXVARCHAR", 16383L),
  ...
)

dolt_type_sizes(types)
```

Arguments

dbObj	the database connection
obj	the data type (vector or data frame)
min_varchar	The minimum size VARCHAR types should be cast as
max_varchar	the maximum size VARCHAR types should be cast as. Larger text data will return types TEXT,MEDIUMTEXT, or LONGTEXT
...	further arguments to methods
types	a character vector of dolt types, e.g., "VARCHAR(12)", "LONGBLOB", "TINYINT", etc.

Details

dolt_type_sizes() takes a vector of SQL types and returns the maximum field size, if applicable.

Value

A character vector of classes, with attributes of the maximum size for text and blob classes

---

dbGetInfo,DoltConnection-method

Get information about a Dolt Database

Description

The Dolt dbGetInfo() returns standard information about a database connection according to the DBI specification, as well as information about the version-control status of the repository such as current branch, last commit, and modified tables. This information is also displayed in the print method for a dolt connection object and in the RStudio connection pane.

Usage

```
## S4 method for signature 'DoltConnection'
dbGetInfo(dbObj, ...)

## S4 method for signature 'DoltConnection'
show(object)
```

Arguments

dbObj	the database connection
...	Other arguments to methods
object	the database connection

See Also

dolt_state dolt_status dolt_last_commit dolt_pane

`dbSendQuery,DoltConnection,character-method`
Miscellaneous Dolt Methods

Description

These methods largely wrap `RMariaDB` methods with small tweaks to work with Dolt databases.

Usage

```
## S4 method for signature 'DoltConnection,character'
dbSendQuery(conn, statement, params = NULL, ...)

## S4 method for signature 'DoltConnection,character'
dbSendStatement(conn, statement, params = NULL, ...)

## S4 method for signature 'DoltResult'
dbClearResult(res, ...)

## S4 method for signature 'DoltConnection'
dbDisconnect(conn, ...)
```

Arguments

`conn`	an DoltConnection object.
`statement`	a character vector of length one specifying the SQL statement that should be executed. Only a single SQL statement should be provided.
`params`	A list of query parameters to be substituted into a parameterized query.
`...`	Unused. Needed for compatibility with generic.#' @export
`res`	A DoltResult object.

`dbWriteTable,DoltConnection,character,data.frame-method`
Write a table to the database

Description

This method uses `dbx::dbxInsert()` as that implementation is much more performant than the standard method from RMariaDB::dbWriteTable(), due to the way Dolt handles repeat `INSERT` statements.

Usage

```
## S4 method for signature 'DoltConnection,character,data.frame'
dbWriteTable(
  conn,
  name,
  value,
  field.types = NULL,
  row.names = FALSE,
  overwrite = FALSE,
  append = FALSE,
  temporary = FALSE,
  batch_size = NULL
)
```

Arguments

conn	a database connection
name	the table name
value	A data frame.
field.types	Optional, overrides default choices of field types, derived from the classes of the columns in the data frame. See dbDataType()
row.names	Either TRUE, FALSE, NA or a string. If TRUE, always translate row names to a column called "row_names". If FALSE, never translate row names. If NA, translate rownames only if they're a character vector. A string is equivalent to TRUE, but allows you to override the default name. For backward compatibility, NULL is equivalent to FALSE.
overwrite	a logical specifying whether to overwrite an existing table or not. Its default is FALSE.
append	a logical specifying whether to append to an existing table in the database If appending, then the table (or temporary table) must exist, otherwise an error is reported. Its default is FALSE.
temporary	If TRUE, creates a temporary table that expires when the connection is closed. For dbRemoveTable(), only temporary tables are considered if this argument is set to TRUE
batch_size	The number of records to insert in a single statement (defaults to all)

Details

This the dependency on dbx may be removed if the base issue is resolved: https://github.com/dolthub/dolt/issues/2091.

See Also

dolt-read

`dolt` *Return a (cached) connection to the default Dolt database*

Description

`dolt()` returns a connection to a default database. It is a convenience wrapper around `dbConnect(dolt_local/remote(), .` that also caches connections for faster loading.

Usage

```
dolt(
  dir = Sys.getenv("DOLT_DIR", "doltdb"),
  dbname = NULL,
  username = Sys.getenv("DOLT_USERNAME", "root"),
  password = Sys.getenv("DOLT_PASSWORD", ""),
  port = Sys.getenv("DOLT_PORT", 3306L),
  host = Sys.getenv("DOLT_HOST", "127.0.0.1"),
  cache_connection = TRUE,
  ...
)
```

Arguments

`dir`	The directory from which to server a `dolt_local()` connection. If `"remote"` a `dolt_remote()` connection will be made and no server will be started.
`dbname`	for remote connections, the database name
`username`	The username. Defaults to "root"
`password`	The login password. Defaults to empty.
`port`	The TCP port for connections. Defaults to 3306.
`host`	The IP of the host. Defaults to the local machine, `127.0.0.1`
`cache_connection`	Should we preserve a cache of the connection? allows faster load times and prevents connection from being garbage-collected.
`...`	further arguments passed to `dolt_server()` or `MariaDB()`

See Also

Other connections: `dolt_local()`, `dolt_remote()`

`dolt-config` *Get and set Dolt configuration variables*

Description

Get and set Dolt configuration variables

Usage

```
dolt_config_get(
  params = NULL,
  global = TRUE,
  local_dir = Sys.getenv("DOLT_DIR")
)

dolt_config_set(params, global = TRUE, local_dir = Sys.getenv("DOLT_DIR"))
```

Arguments

`params`	What parameters to get or set. Can include `user.name`, `user.email`, and `user.creds`. For `dolt_config_set`, this should be a named character vector or list with parameter names and values.
`global`	Set global or database-specific credentials
`local_dir`	if not `global`, what local database to set variables for

See Also

dolt_vars

`dolt-read` *Reading from a Dolt database.*

Description

These methods are extensions of standard DBI functions such as DBI::dbReadTable. They differ in that they can take an `as_of` argument, reading historical data from the database that was written as of a certain date or commit hash, or from a different branch.

Usage

```
## S4 method for signature 'DoltConnection,character'
dbReadTable(
  conn,
  name,
  as_of = NULL,
  ...,
  row.names = FALSE,
  check.names = TRUE
)

## S4 method for signature 'DoltConnection'
dbListTables(conn, as_of = NULL, ...)

## S4 method for signature 'DoltConnection'
dbListObjects(conn, prefix = NULL, as_of = NULL, ...)

## S4 method for signature 'DoltConnection,character'
dbExistsTable(conn, name, as_of = NULL, ...)
```

Arguments

conn	a dolt connection object, produced by DBI::dbConnect() or dolt()
name	a character string specifying a table name.
as_of	A dolt commit hash, branch name, or object coercible to POSIXct
...	Unused, needed for compatibility with generic.
row.names	Either TRUE, FALSE, NA or a string. If TRUE, always translate row names to a column called "row_names". If FALSE, never translate row names. If NA, translate rownames only if they're a character vector. A string is equivalent to TRUE, but allows you to override the default name. For backward compatibility, NULL is equivalent to FALSE.
check.names	If TRUE, the default, column names will be converted to valid R identifiers.
prefix	A fully qualified path in the database's namespace, or NULL. This argument will be processed with dbUnquoteIdentifier(). If given the method will return all objects accessible through this prefix.

Value

A data.frame in the case of dbReadTable(); a character vector of names for dbListTables() and dbListObjects(), and a logical result for dbExistsTable().

See Also

Querying Historical Data with AS OF Queries on the DoltHub blog, and RMariaDB methods upon which these are built.

dolt-vars *Configuration variable options*

Description

The doltr package's behavior can be modified by setting these environment variables:

Details

- DOLT_DIR set the default directory to look for a dolt database and run a server when using dolt_local() and dolt(). Defaults to "doltdb".
- DOLT_PORT sets the port to connect to or to run the server on. Defaults to 3306.
- DOLT_HOST sets the host IP to connect to or to run the server on. Defaults to 127.0.0.1.
- DOLT_CONFIG_FILE is the path to a file with additional configuration options for the dolt sql server. See https://docs.dolthub.com/interfaces/cli#dolt-sql-server for options.
- DOLT_PATH specifies the path to the dolt binary if running locally. Defaults to the one found in the system path.
- DOLT_COLLECT specifies whether dolt convenience functions returning data should return fully collected tibbles or lazy tibbles for further processing. Set it to 0 or `false` to disable, potentially for when large databases with long histories yield very large responses to commands like `dolt_log()` or `dolt_diffs()`.
- DOLT_VERBOSE will print the SQL or command-line statements executed when running functions that wrap database or system calls. Useful for understanding how dolt commands work. Set to 1 or `true` to enable this behavior.
- DOLT_WATCH determines whether the RStudio Connection pane automatically updates in response to changes in the database. Set it to 0 or `false` to disable this behavior.
- DOLT_ROOT_DIR the directory where Dolt global configuration and credential data is stored (`~/.dolt` by default). Note this can also be set in your shell to configure command-line dolt.

See Also

dolt-config

dolt_add *Add, commit, and reset tables in a dolt database*

Description

Add, commit, and reset tables in a dolt database

Usage

```
dolt_add(tables = NULL, conn = dolt(), collect = NULL, show_sql = NULL)

dolt_commit(
  all = TRUE,
  message = NULL,
  author = NULL,
  date = NULL,
  allow_empty = FALSE,
  conn = dolt(),
  collect = NULL,
  show_sql = NULL
)

dolt_reset(
  hard = FALSE,
  tables = NULL,
  conn = dolt(),
  collect = NULL,
  show_sql = NULL
)
```

Arguments

tables	Which tables to be reset? Defaults to all tables if NULL.
conn	the database connection
collect	whether to collect the result into R or return a dbplyr::tbl_lazy() to be further processed before collecting. Defaults to TRUE, can be set with the environment variable DOLT_COLLECT.
show_sql	Whether to print the SQL statements used internally to fetch the data. Useful for learning how Dolt works internally. Defaults to FALSE, can be set with the environment variable DOLT_VERBOSE.
all	stage all tables before committing?
message	A commit message. If NULL in an interactive session, the user will be prompted. Otherwise will error if empty.
author, date	Author and date. If null, uses the ones set in dolt-config. Author should be in the format "A U Thor author@example.com"
allow_empty	Allow recording a commit that has the exact same data as its sole parent. This is usually a mistake, so it is FALSE by default.
hard	Reset working and staged tables? If FALSE (default), a "soft" reset will be performed, only unstaging staged tables. If TRUE, all working and staged changes will be discarded.

dolt_branches *Dolt System Tables*

Description

These functions query the dolt database for system tables that describe the database version history and structure.

Usage

```
dolt_branches(conn = dolt(), collect = NULL, show_sql = NULL)

dolt_remotes(conn = dolt(), collect = NULL, show_sql = NULL)

dolt_docs(conn = dolt(), collect = NULL, show_sql = NULL)

dolt_log(conn = dolt(), collect = NULL, show_sql = NULL)
```

Arguments

conn	the database connection
collect	whether to collect the result into R or return a `dbplyr::tbl_lazy()` to be further processed before collecting. Defaults to `TRUE`, can be set with the environment variable `DOLT_COLLECT`.
show_sql	Whether to print the SQL statements used internally to fetch the data. Useful for learning how Dolt works internally. Defaults to `FALSE`, can be set with the environment variable `DOLT_VERBOSE`.

dolt_checkout *Navigate dolt history*

Description

`dolt_checkout()` checks out a dolt branch, setting that branch as HEAD and bringing you to its tip. `dolt_use()` sets the database to use a specific commit as it's state and puts you in read-only mode.

Usage

```
dolt_checkout(
  branch,
  b = FALSE,
  start_point = NULL,
  conn = dolt(),
  collect = NULL,
```

```
  show_sql = NULL
)

dolt_use(hash = NULL, conn = dolt())
```

Arguments

branch	the branch to check out
b	whether to create a new branch
start_point	a commit hash from which the branch should start. If NULL, starts from current HEAD.
conn	the database connection
collect	whether to collect the result into R or return a dbplyr::tbl_lazy() to be further processed before collecting. Defaults to TRUE, can be set with the environment variable DOLT_COLLECT.
show_sql	Whether to print the SQL statements used internally to fetch the data. Useful for learning how Dolt works internally. Defaults to FALSE, can be set with the environment variable DOLT_VERBOSE.
hash	the commit hash you want to set the database to. If NULL, checks out the head of the main branch and brings you out of read-only mode.

---

dolt_diffs *Examine information about dolt tables and diffs*

Description

Examine information about dolt tables and diffs

Usage

```
dolt_diffs(table, to, from, conn = dolt(), collect = NULL, show_sql = NULL)

dolt_table_history(table, conn = dolt(), collect = NULL, show_sql = NULL)
```

Arguments

table	character the name of a table in the database
to	commit to compare to
from	commit to compare from
conn	the database connection
collect	whether to collect the result into R or return a dbplyr::tbl_lazy() to be further processed before collecting. Defaults to TRUE, can be set with the environment variable DOLT_COLLECT.
show_sql	Whether to print the SQL statements used internally to fetch the data. Useful for learning how Dolt works internally. Defaults to FALSE, can be set with the environment variable DOLT_VERBOSE.

dolt_dump *Export data from a dolt database*

Description

Export data from a dolt database

Usage

```
dolt_dump(
  format = c("sql", "csv", "json", "parquet"),
  out = NULL,
  overwrite = FALSE,
  dir = Sys.getenv("DOLT_DIR", "doltdb")
)
```

Arguments

format	the export data format. One of "sql", "csv", "json", or "parquet"
out	the location on-disk for export. In the case of "sql", format, a single file path (default doltdump.sql), otherwise a directory for all tables to be dumped as separate files (default "doltdump")
overwrite	whether to overwrite existing files/directories.
dir	path to dolt database on-disk

Value

the path(s) of exported files

dolt_init *Initiate a dolt database directory*

Description

Initiate a dolt database directory

Usage

```
dolt_init(dir = Sys.getenv("DOLT_DIR", "doltdb"))
```

Arguments

dir	path to the directory. Will be created if it does not exist

dolt_local *Connect to a local dolt database directory*

Description

dolt_local() creates a DoltLocalDriver, which can generate a DoltLocalConnection. Unlike dolt_remote() and DoltDriver, *local* connections are for dolt databases stored in directories on-disk, and take a directory name as an argument. The local connection type starts and manages a dolt SQL server in the background serving that directory, connects to it and returns the connection. Parameters govern both the server and connection

Local dolt connection objects contain additional slots including the database path on-disk and an external pointer to the server process, and these are returned via dbGetInfo and displayed in the connection print method. The dbDisconnect method kills the background server if no other processes are connected to it.

Multi-user or other, more complicated networking set-ups should use dolt_server() and dolt_remote() directly.

Usage

```
dolt_local()

## S4 method for signature 'DoltLocalDriver'
dbUnloadDriver(drv, ...)

## S4 method for signature 'DoltLocalDriver'
show(object)

## S4 method for signature 'DoltLocalDriver'
dbConnect(
  drv,
  dir = Sys.getenv("DOLT_DIR", "doltdb"),
  username = Sys.getenv("DOLT_USERNAME", "root"),
  password = Sys.getenv("DOLT_PASSWORD", ""),
  port = Sys.getenv("DOLT_PORT", 3306L),
  host = Sys.getenv("DOLT_HOST", "127.0.0.1"),
  find_port = TRUE,
  find_server = TRUE,
  autocommit = TRUE,
  server_args = list(),
  ...
)

## S4 method for signature 'DoltLocalConnection'
dbGetInfo(dbObj, ...)

## S4 method for signature 'DoltLocalConnection'
```

```
show(object)

## S4 method for signature 'DoltLocalConnection'
dbDisconnect(conn, ...)

## S4 method for signature 'DoltLocalConnection'
dbIsValid(dbObj, ...)
```

Arguments

drv	an object of class DoltLocalDriver, created by dolt_local().
...	additional arguments to pass to RMariaDB
object	a connection object
dir	The dolt directory to serve and connect to
username	The username. Defaults to "root"
password	The login password. Defaults to empty.
port	The TCP port for connections. Defaults to 3306.
host	The IP of the host. Defaults to the local machine, 127.0.0.1
find_port	whether to find an open port if the default is used by another process
find_server	whether to look for another server process serving the same directory before creating a new one
autocommit	Whether to autocommit changes in the *SQL* sense. That is, to flush pending changes to disk and update the working set.
server_args	a list of additional arguments to pass to dolt_server()
dbObj	the database connection
conn	the database connection

See Also

Other connections: dolt_remote(), dolt()

---

dolt_pane *Open a Dolt connection pane in RStudio*

Description

This function launches the RStudio "Connection" pane to interactively explore the database. The pane will show the database versioning state, tables stored in the database, and dolt system tables showing history.

Usage

```
dolt_pane(conn = dolt())

update_dolt_pane(conn = dolt())

close_dolt_pane(conn = dolt())
```

Arguments

conn	a dolt connection. If a path is provided instead, a connection will be created to the path using `dolt()`.

Details

When running dolt interactively, the connection pane will automatically update in response to most queries that modify the database state. You can stop this behavior by setting the `DOLT_WATCH` environment variable to `0` or `false`. See dolt_vars for more configuration variables

Value

The connection object (invisibly)

---

dolt_push *Work with dolt repository remotes*

Description

Work with dolt repository remotes

Usage

```
dolt_push(
  remote = NULL,
  remote_branch = NULL,
  ref = NULL,
  set_upstream = FALSE,
  force = FALSE,
  conn = dolt(),
  collect = NULL,
  show_sql = NULL
)

dolt_pull(
  remote = NULL,
  squash = FALSE,
  conn = dolt(),
  collect = NULL,
```

```
  show_sql = NULL
)

dolt_fetch(
  remote = NULL,
  ref = FALSE,
  force = FALSE,
  conn = dolt(),
  collect = NULL,
  show_sql = NULL
)

dolt_clone(
  remote_url,
  remote = "origin",
  new_dir = basename(remote_url),
  branch = NULL
)
```

Arguments

remote	the name of the remote. "origin" is used by default
remote_branch	the name of the remote branch to use with set_upstream. Current local branch is used by default
ref	the branch reference
set_upstream	whether to set the remote branch reference to track
force	whether to overwrite any conflicting history the current branch
conn	the database connection
collect	whether to collect the result into R or return a `dbplyr::tbl_lazy()` to be further processed before collecting. Defaults to `TRUE`, can be set with the environment variable `DOLT_COLLECT`.
show_sql	Whether to print the SQL statements used internally to fetch the data. Useful for learning how Dolt works internally. Defaults to `FALSE`, can be set with the environment variable `DOLT_VERBOSE`.
squash	whether to merge changes to the working set without updating the commit history
remote_url	the remote URL to clone
new_dir	the directory to clone into
branch	the branch to clone. If NULL, clones all branches

dolt_remote *Connect to a dolt database*

Description

`dolt_remote()` is a DBI Driver to connect to a remote dolt server via a port. It, `DoltDriver`, and `DoltConnection` class are wrappers around the around classes and methods from the `RMariaDB` package.

Most parameters can be specified with package configuration environment variables.

Usage

```
dolt_remote()

## S4 method for signature 'DoltDriver'
dbUnloadDriver(drv, ...)

## S4 method for signature 'DoltDriver'
show(object)

## S4 method for signature 'DoltDriver'
dbConnect(
  drv = dolt_remote(),
  dbname = Sys.getenv("DOLT_DIR", "doltdb"),
  username = Sys.getenv("DOLT_USERNAME", "root"),
  password = Sys.getenv("DOLT_PASSWORD", ""),
  host = Sys.getenv("DOLT_HOST", "127.0.0.1"),
  port = Sys.getenv("DOLT_PORT", 3306L),
  autocommit = TRUE,
  ...
)
```

Arguments

`drv`	an object of class `DoltDriver`, created by `dolt_remote()`.
`...`	other arguments passed to RMariaDB::MariaDB
`object`	a connection object
`dbname`	The database name
`username`	The username. Defaults to "root"
`password`	The login password. Defaults to empty.
`host`	The IP of the host. Defaults to the local machine, `127.0.0.1`
`port`	The TCP port for connections. Defaults to 3306.
`autocommit`	Whether to autocommit changes in the *SQL* sense. That is, to flush pending changes to disk and update the working set.

Details

Most methods fall back to those for RMariaDB.

See Also

Other connections: dolt_local(), dolt()

`dolt_server` *Start up a dolt SQL server and return the server process handle*

Description

Start up a dolt SQL server and return the server process handle

Usage

```
dolt_server(
  dir = Sys.getenv("DOLT_DIR", "doltdb"),
  username = Sys.getenv("DOLT_USERNAME", "root"),
  password = Sys.getenv("DOLT_PASSWORD", ""),
  port = Sys.getenv("DOLT_PORT", 3306L),
  host = Sys.getenv("DOLT_HOST", "127.0.0.1"),
  find_port = TRUE,
  find_server = TRUE,
  multi_db = FALSE,
  autocommit = TRUE,
  read_only = FALSE,
  log_level = "info",
  log_out = NULL,
  timeout = 28800000,
  query_parallelism = 2,
  max_connections = 100,
  config_file = Sys.getenv("DOLT_CONFIG_FILE", "")
)
```

Arguments

`dir`	The dolt directory to serve
`username`	The username. Defaults to "root"
`password`	The login password. Defaults to empty.
`port`	The TCP port for connections. Defaults to 3306.
`host`	The IP of the host. Defaults to the local machine, `127.0.0.1`
`find_port`	if TRUE, switch to a different port if `port` is used by another process

find_server	if TRUE, find a server process serving the same directory rather than starting a new one. Note that other server options will be ignored. This allows the server to be used across R sessions. Note that to make best use of this you may want to turn off the "Quit child processes on exit" option in RStudio project options.
multi_db	Serve multiple databases? If TRUE, dir should be a directory with multiple subdirectories that are dolt databases
autocommit	Automatically commit database changes to the working set? If FALSE, anything not manually committed will be lost.
read_only	should the database only allow read_only connections?
log_level	Defines the level of logging provided. Options are "trace", debug", "info", "warning", "error", and "fatal" (default "info").
log_out	Where logging output should be directed. If "|" it is passed to std_out(), if NULL (default), it is suppressed. Can also take a filename. See processx::run().
timeout	Defines the timeout, in seconds, used for connections (default 28800000)
query_parallelism	Set the number of go routines spawned to handle each query (default 2)
max_connections	Set the number of connections handled by the server (default 100)
config_file	The path to a YAML config file to set these and additional server configuration values. See options in the dolt documentation.

Value

A dolt_server object that is also a ps::ps_handle()

---

dolt_state *Get information about a dolt database*

Description

These functions yield information about the current state of a dolt database. dolt_state() provides information on current branch or headless commit. dolt_status() summarizes changes to the database in working or staged tables (from the dolt_status table). dolt_last_commit() pulls the most recent value from the dolt_log table. All have pretty-print methods for the objects returned but can be interrogated for more detail.

Usage

```
dolt_state(conn = dolt())

dolt_status(conn = dolt())

dolt_last_commit(conn = dolt())
```

Arguments

conn	the database connection

Details

Values from each of these functions are returned as part of the `dbGetInfo()` method and are part of the information shown in the `DoltConnection` print method and in the RStudio Connection pane for a Dolt Database.

Value

A data frame of class "dolt_status" and tibble::tbl_df. It pretty-prints as an abbreviated summary of status.

`is_dolt_installed` *Find and check for the presence of a dolt binary*

Description

Find and check for the presence of a dolt binary

Usage

```
is_dolt_installed()

dolt_version()

dolt_path()
```

Index