Package ‘RPresto’

October 18, 2019

Title  DBI Connector to Presto
Version 1.3.4
Copyright Facebook, Inc. 2015-present.
Description Implements a 'DBI' compliant interface to Presto. Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes: <https://prestodb.io/>.
Depends R (>= 3.1.0), methods
Imports DBI (>= 0.3.0), httr (>= 0.6), openssl, jsonlite, stringi, stats, Rcpp (>= 0.12.7), utils
Suggests testthat, dplyr (>= 0.7.0), dbplyr
License BSD_3_clause + file LICENSE
URL https://github.com/prestodb/RPresto
BugReports https://github.com/prestodb/RPresto/issues
Encoding UTF-8
LazyData true
Collate 'PrestoDriver.R' 'Presto.R' 'utility_functions.R'

'PrestoSession.R' 'PrestoConnection.R' 'PrestoCursor.R'

'PrestoResult.R' 'RcppExports.R' 'src.presto.R'

'copy.to.src.presto.R' 'db.data.type.PrestoConnection.R'

'dbplyr_compatible.R' 'db.explain.PrestoConnection.R'

'db.query.rows.PrestoConnection.R' 'request_headers.R'

'dbHasCompleted.R' 'dbClearResult.R' 'dbConnect.R'

'dbDataType.R' 'dbDisconnect.R' 'dbSendQuery.R'

'json.tabular.to.data.frame.R' 'extract.data.R'

'parse.response.R' 'dbFetch.R' 'dbGetQuery.R' 'dbListTables.R'

'dbExistsTable.R' 'dbGetInfo.R' 'dbGetRowCount.R'

'dbGetStatement.R' 'dbIsValid.R' 'dbListFields.R'

'dbUnloadDriver.R' 'description_from_info.R'

'db_desc.PrestoConnection.R' 'db_query_fields.R' 'fetch.R'

'help.R' 'sql_translate_env.R' 'tbl.src.presto.R'

RoxygenNote 6.1.1
Description
Return the corresponding presto data type for the given R object

Usage
```
## S4 method for signature 'PrestoDriver'
dbDataType(dbObj, obj, ...)
```

Arguments
- `dbObj` A `PrestoDriver` object
- `obj` Any R object
- `...` Extra optional parameters, not currently used

Details
The default value for unknown classes is 'VARCHAR'.
'ARRAY's and 'MAP's are supported with some caveats. Unnamed lists will be treated as 'ARRAY's and named lists will be a 'MAP'. All items are expected to be of the same corresponding Presto type, otherwise the default 'VARCHAR' value is returned. The key type for 'MAP's is always 'VARCHAR'. The 'value' type for empty lists is always a 'VARCHAR'.
Value

A character value corresponding to the Presto type for obj

Examples

drv <- RPresto::Presto()
dbDataType(drv, list())
dbDataType(drv, 1)
dbDataType(drv, NULL)
dbDataType(drv, list(list(a=Sys.Date())))
dbDataType(drv, as.POSIXct('2015-03-01 00:00:00', tz='UTC'))
dbDataType(drv, Sys.time())
# Data types for ARRAY or MAP values can be tricky
all.equal('VARCHAR', dbDataType(drv, list(1, 2, 3L)))

Description

Metadata about database objects

For the PrestoResult object, the implementation returns the additional stats field which can be used to implement things like progress bars. See the examples section.

Usage

## S4 method for signature 'PrestoDriver'
dbGetInfo(dbObj)

## S4 method for signature 'PrestoConnection'
dbGetInfo(dbObj)

## S4 method for signature 'PrestoResult'
dbGetInfo(dbObj)

Arguments

dbObj A PrestoDriver, PrestoConnection or PrestoResult object

Value

PrestoResult A list with elements

- **statement** The SQL sent to the database
- **row.count** Number of rows fetched so far
- **has.completed** Whether all data has been fetched
- **stats** Current stats on the query
Examples

```r
## Not run:
conn <- dbConnect(Presto(), 'localhost', 7777, 'onur', 'datascience')
result <- dbSendQuery(conn, 'SELECT * FROM jonchang_iris')
iris <- data.frame()
progress.bar <- NULL
while (!dbHasCompleted(result)) {
  chunk <- dbFetch(result)
  if (!NROW(iris)) {
    iris <- chunk
  } else if (NROW(chunk)) {
    iris <- rbind(iris, chunk)
  }
  stats <- dbGetInfo(result)[['stats']]
  if (is.null(progress.bar)) {
    progress.bar <- txtProgressBar(0, stats[['totalSplits']], style=3)
  } else {
    setTxtProgressBar(progress.bar, stats[['completedSplits']])
  }
}
close(progress.bar)
## End(Not run)
```

---

Presto

**Connect to a Presto database**

Description

Connect to a Presto database

Usage

```r
Presto(...)
```

## S4 method for signature 'PrestoDriver'
```r
dbConnect(drv, catalog, schema, user,
          host = "localhost", port = 8080, source = getPackageName(),
          session.timezone = "UTC", parameters = list(), ...)
```

## S4 method for signature 'PrestoConnection'
```r
dbDisconnect(conn)
```

Arguments

<table>
<thead>
<tr>
<th>...</th>
<th>currently ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>drv</td>
<td>A driver object generated by <code>Presto</code></td>
</tr>
<tr>
<td>catalog</td>
<td>The catalog to be used</td>
</tr>
</tbody>
</table>
PrestoCursor-class

schema The schema to be used
user The current user
host The presto host to connect to
port Port to use for the connection
source Source to specify for the connection
session.timezone Time zone to use for the connection. Presto returns timestamps without time zones with respect to this value. The time arithmetic (e.g. adding hours) will also be done in the given time zone. See the session.timezone tests for examples.
parameters A list of extra parameters to be passed in the ‘X-Presto-Session’ header
cconn A PrestoConnection object

Value

Presto A PrestoDriver object

[dbConnect] A PrestoConnection object

[dbDisconnect] A logical value indicating success

Examples

## Not run:
```r
conn <- dbConnect(Presto(), catalog = 'hive', schema = 'default',
                   user = 'onur', host = 'localhost', port = 8080,
                   session.timezone='US/Eastern')
dbListTables(conn, 'K_iris')
dbDisconnect(conn)
```

## End(Not run)

---

PrestoCursor-class Internal implementation detail class needed for its side-effects. When dbFetch is called, we need to both return the data and update the uri to the next value.

Description

Internal implementation detail class needed for its side-effects. When dbFetch is called, we need to both return the data and update the uri to the next value.
PrestoSession-class  
*Internal implementation detail class needed for its side-effects. When SET/RESET SESSION queries are called, session parameters need to be maintained by the client and requires an in-place update.*

Description

Internal implementation detail class needed for its side-effects. When SET/RESET SESSION queries are called, session parameters need to be maintained by the client and requires an in-place update.

R Presto  
*R Presto*

Description

A DBI-compliant interface to Presto.

Examples

```r
library("DBI")
library("R Presto")
```

src_presto  
*dplyr integration to connect to a Presto database.*

Description

Allows you to connect to an existing database through a presto connection.

Use `src_presto` to connect to an existing database, and `tbl` to connect to tables within that database. If you're unsure of the arguments to pass, please ask your database administrator for the values of these variables.

Usage

```
src_presto(catalog = NULL, schema = NULL, user = NULL, host = NULL, 
port = NULL, source = NULL, session.timezone = NULL, 
parameters = NULL, ...)
```

```
tbl.src_presto(src, from, ...)
```
Arguments

catalog  Catalog to use in the connection
schema   Schema to use in the connection
user     User name to use in the connection
host     Host name to connect to the database
port     Port number to use with the host name
source   Source to specify for the connection
session.timezone  Time zone for the connection
parameters Additional parameters to pass to the connection
... For src_presto other arguments passed on to the underlying database connector dbConnect. For tbl.src_presto, it is included for compatibility with the generic, but otherwise ignored.
src A presto src created with src_presto.
from Either a string giving the name of table in database, or sql described a derived table or compound join.

Examples

```r
## Not run:
# To connect to a database
my_db <- src_presto(catalog = "hive", schema = "web", user = "onur",
                     host = "localhost", port = 8888, session.timezone='Asia/Kathmandu')

## End(Not run)
## Not run:
First create a database connection with src_presto, then reference a tbl within that database
my_tbl <- tbl(my_db, "my_table")

## End(Not run)
```
Index

dbConnect,PrestoDriver-method (Presto), 4
dbDataType,PrestoDriver-method, 2
dbDisconnect,PrestoConnection-method (Presto), 4
dbGetInfo,PrestoConnection-method (dbGetInfo,PrestoDriver-method), 3
dbGetInfo,PrestoDriver-method, 3
dbGetInfo,PrestoResult-method (dbGetInfo,PrestoDriver-method), 3

list, 3, 5
logical, 5

package-RPresto (RPresto), 6
Presto, 4, 4
PrestoConnection, 3, 5
PrestoCursor (PrestoCursor-class), 5
PrestoCursor-class, 5
PrestoDriver, 2, 3, 5
PrestoResult, 3
PrestoSession (PrestoSession-class), 6
PrestoSession-class, 6

RPresto, 6
RPresto-package (RPresto), 6

sql, 7
src_presto, 6

tbl.src_presto (src_presto), 6