Package 'influxdbclient'

July 2, 2021

Title R Package for InfluxDB 2.x client

Version 0.1.0

Description R Package for InfluxDB 2.x client.

URL https://github.com/bonitoo-io/influxdb-client-r

BugReports https://github.com/bonitoo-io/influxdb-client-r/issues

Depends R (>= 3.3)

Encoding UTF-8

License MIT License + file LICENSE

LazyData true

Imports R6, httr, jsonlite, bit64, nanotime (>= 0.3.0), plyr

RoxygenNote 7.1.1

Suggests testthat, httptest

R topics documented:

InfluxDBClient   *InfluxDBClient*

Description

Client for querying from and writing to InfluxDB 2.x.

Format

An `R6Class` object

1

Public fields

`url` Database URL

`token` Authentication token

`org` Organization name

`dialect` Flux dialect

`retryOptions` Retry options

Methods

Public methods:

- [`InfluxDBClient$new()`](#)
- [`InfluxDBClient$health()`](#)
- [`InfluxDBClient$query()`](#)
- [`InfluxDBClient$ready()`](#)
- [`InfluxDBClient$write()`](#)
- [`InfluxDBClient$as.lp()`](#)
- [`InfluxDBClient$clone()`](#)

Method `new()`: Creates instance of `InfluxDBClient`.

Usage:

`InfluxDBClient$new(url, token, org, retryOptions = NULL)`

Arguments:

`url` InfluxDB instance URL

`token` Authentication token

`org` Organization name

`org` Retry options. See `RetryOptions` for details. Set to `TRUE` for default retry options. Default is `NULL` which disables retries.

Method `health()`: Gets health info of the InfluxDB instance.

Usage:

`InfluxDBClient$health()`

Returns: Named list with name, `message`, `status`, `version`, `commit` elements or error

Method `query()`: Queries data in the InfluxDB instance.

Usage:

`InfluxDBClient$query(text, POSIXctCol = c(`_time` = "time"))`

Arguments:

`text` Flux query

`POSIXctCol` Flux time to (new) POSIXct column mapping (named list). Default is `c("_time"="time")`. Use `NULL` to skip it.

Returns: Data as (list of) `data.frame`

Method `ready()`: Gets readiness status of the InfluxDB instance.

Usage:
```
InfluxDBClient$ready()
```

Returns: Named list with `status`, `started` and `up` elements or error

Method `write()`: Writes data to the InfluxDB instance.

Usage:
```
InfluxDBClient$write(
  x,
  bucket,
  batchSize = 5000,
  precision = c("ns", "us", "ms", "s"),
  measurementCol = "_measurement",
  tagCols = NULL,
  fieldCols = c(`_field` = "_value"),
  timeCol = "_time",
  ...
)
```

Arguments:

`x` Data as (list of) `data.frame`

`bucket` Target bucket name

`batchSize` Batch size. Positive number or `FALSE` to disable. Default is `5000`.

`precision` Time precision

`measurementCol` Name of measurement column. Default is `"_measurement"`.

`tagCols` Names of tag (index) columns

`fieldCols` Names of field columns. In case of unpivoted data previously retrieved from InfluxDB, use default value ie. named list `c("_field"="_value")`. For all other cases, just use simple vector of column names (see Examples).

`timeCol` Name of time column. The column values should be either of `nanotime` or `POSIXct` type. Default is `"_time"`.

Method `as.lp()`: Coerces data to InfluxDB line protocol. *For debugging and introspection purposes. Parameters are the same as for* `write` *method.*

Usage:
```
InfluxDBClient$as.lp(
  x,
  precision = c("ns", "us", "ms", "s"),
  measurementCol = "_measurement",
  tagCols = NULL,
  fieldCols = c(`_field` = "_value"),
  timeCol = "_time"
)
```

Returns: line protocol text

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
    InfluxDBClient$clone(deep = FALSE)
```
Arguments:

deep Whether to make a deep clone.

Examples

```
## Not run:
# Instantiation
client <- InfluxDBClient$new(url = "http://localhost:8086",
                            token = "my-token",
                            org = "my-org")

# Query
data <- client$query('from(bucket: "my-bucket") |> range(start: -1h) |> drop(columns: ["_start", "_stop"])')

# Write
data <- data.frame(...)
client$write(data, bucket = "my-bucket", precision = "us",
             measurementCol = "name",
             tagCols = c("location", "id"),
             fieldCols = c("altitude", "temperature"),
             timeCol = "time")

# Ready status
ready <- client$ready()

# Healt info
ready <- client$health()

## End(Not run)
```

RetryOptions   *RetryOptions*

Description

Retry options may be specified as optional argument to `write`.

Format

An `R6Class` object

Public fields

`retryJitter` Maximum number of random milliseconds included in delay. Default is 0.

`retryInterval` First retry delay in seconds. Default is 5.

`maxDelay` Maximum delay between retries in seconds. Default is 125.

`maxRetryTime` Maximum time to spend retrying in seconds. Default is 180.

`maxAttempts` Number of retry attempts. Default is 5.

`exponentialBase` Base for exponential backoff strategy. Default is 2.

Methods

    Public methods:

- `RetryOptions$new()`
- `RetryOptions$clone()`

Method `new()`: Creates instance of `RetryOptions`.

Usage:

```
RetryOptions$new(
  retryJitter = 0,
  retryInterval = 5,
  maxDelay = 125,
  maxRetryTime = 180,
  maxAttempts = 5,
  exponentialBase = 2,
  ...
)
```

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
RetryOptions$clone(deep = FALSE)
```

Arguments:

deep   Whether to make a deep clone.

Index

InfluxDBClient, 1

RetryOptions, 4