Package 'r2time'

July 3, 2014

Type Package

Title Analysis of Large opentsdb time-series datasets in HBase

Version 1.0

Date 2013-07-16

Author Bikash Agrawal

Maintainer Bikash Agrawal <er.bikash21@gmail.com>

Description Analysis of Large opentsdb time-series datasets in HBase

License GNU

R topics documented:

`r2time-package` *Analysis of Large opentsdb time-series datasets in HBase*

Description

Analysis of Large opentsdb time-series datasets in HBase

Details

Package:	r2time
Type:	Package
Version:	1.0
Date:	2013-07-16

~~ An overview of how to use the package, including the most important ~~ ~~ functions ~~

Author(s)

Bikash Agrawal

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

`pushdata` *initilization function*

Description

initilization function

Usage

```
pushdata(l, x)
```

Arguments

`l`

`x`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (l, x)
{
    assign(l, append(eval(as.name(l)), x), envir = parent.frame())
  }
```

---

pushList *initilization function*

Description

initilization function

Usage

```
pushList(l)
```

Arguments

```
l
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (l)
{
    lst <- get(l, parent.frame())
    a <- c("1")
    y <- append(eval(as.name("a")), lst)
    assign(l, y, envir = parent.frame())
  }
```

---

r2t.byte2long *initilization function*

Description

initilization function

Usage

```
r2t.byte2long(ra, data)
```

Arguments

ra

data

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, data)
{
    res <- .jcall(ra, "J", "byte2long", data)
  }
```

---

r2t.Bytes2Float *initilization function*

Description

initilization function

Usage

```
r2t.Bytes2Float(data)
```

Arguments

data

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (data)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "F", "Bytes2Float", data)
  }
```

r2t.bytes2float *initilization function*

Description

initilization function

Usage

```
r2t.bytes2float(value)
```

Arguments

```
value
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (value)
{
    obj <- .jnew("org.apache.hadoop.hbase.util.Bytes")
    res <- .jcall(obj, "F", "toFloat", value)
  }
```

r2t.connect *initilization function*

Description

initilization function

Usage

```
r2t.connect(ra, tablename)
```

Arguments

```
ra
```

```
tablename
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, tablename)
{
    tb <- .jnew("org/apache/hadoop/hbase/client/HTable", tablename)
    j <- function(ra, tb) {
        return(list(getTagK = function() getTagK(ra = ra, tb = tb),
            getTagV = function() getTagV(ra = ra, tb = tb), table = tb))
    }
    return(j(ra, tb))
  }
```

`r2t.connect` *initilization function*

Description

initilization function

Usage

```
r2t.connect(ra, tablename)
```

Arguments

`ra`

`tablename`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, tablename)
{
    tb <- .jnew("org/apache/hadoop/hbase/client/HTable", tablename)
    j <- function(ra, tb) {
        return(list(getTagK = function() getTagK(ra = ra, tb = tb),
            getTagV = function() getTagV(ra = ra, tb = tb), table = tb))
    }
    return(j(ra, tb))
  }
```

r2t.convertByteArraytoFloat *initilization function*

Description

initilization function

Usage

```
r2t.convertByteArraytoFloat(v)
```

Arguments

v

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (v)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[F", "convertBytetoFloat", v)
  }
```

r2t.getBaseTimestamp *initilization function*

Description

initilization function

Usage

```
r2t.getRealTimestamp(basetimestamp,delta)
```

Arguments

rowkey

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (rowkey)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "I", "getBaseTimestamp", rowkey)
  }
```

---

r2t.getLong *initilization function*

Description

initilization function

Usage

```
r2t.getLong(ra, data, offset)
```

Arguments

ra

data

offset

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, data, offset)
{
    res <- .jcall(ra, "J", "getLong", data, offset)
  }
```

r2t.getMetrics *initilization function*

Description

initilization function

Usage

```
r2t.getMetrics(ra, tb)
```

Arguments

ra

tb

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, tb)
{
    res <- .jcall(ra, "[B", "getMetrics", tb)
  }
```

r2t.getRowKey *initilization function*

Description

initilization function

Usage

```
r2t.getRowBaseTimestamp(x)
```

Arguments

sdate

edate

tagk

tagv

metrics

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (sdate, edate, tagk, tagv, metrics)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[J", "getRowKey", sdate, edate, tagk,
        tagv, metrics)
  }
```

`r2t.getRowKey` *initilization function*

Description

initilization function

Usage

```
r2t.getRowKey(sdate, edate, tagk, tagv, metrics)
```

Arguments

sdate
edate
tagk
tagv
metrics

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (sdate, edate, tagk, tagv, metrics)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[J", "getRowKey", sdate, edate, tagk,
        tagv, metrics)
  }
```

r2t.getRowkeyFilter *initilization function*

Description

initilization function

Usage

```
r2t.getRowkeyFilter(sdate, edate, metrics, tagk, tagv)
```

Arguments

sdate

edate

metrics

tagk

tagv

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (sdate, edate, metrics, tagk, tagv)
{
    tagk = pushList("tagk")
    if (!is.array(tagk))
        tagkey = array(data = tagk, dim = length(tagk))
    else tagkey = tagk
    tagv = pushList("tagv")
    if (!is.array(tagv))
        tagvalue = array(data = tagv, dim = length(tagv))
    else tagvalue = tagv
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[S", "getRowkeyFilter", sdate, edate,
        metrics, tagkey, tagvalue)
  }
```

r2t.getTagKeys *initilization function*

Description

initilization function

Usage

```
r2t.getTagKeys(ra, tb)
```

Arguments

ra

tb

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, tb)
{
    res <- .jcall(ra, "[B", "getTagK", tb)
  }
```

---

r2t.getTagValue *initilization function*

Description

initilization function

Usage

```
r2t.getTagValue(ra, tb, tag)
```

Arguments

ra

tb

tag

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, tb, tag)
{
    res <- .jcall(ra, "[B", "getTagv", tb)
  }
```

r2t.getTimeSereiesData *initilization function*

Description

initilization function

Usage

```
r2t.getTimeSereiesData(sdate, edate, tagk, tagv, metrics)
```

Arguments

sdate

edate

tagk

tagv

metrics

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (sdate, edate, tagk, tagv, metrics)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[[J", "getTimeSereiesData", sdate, edate,
        tagk, tagv, metrics)
  }
```

r2t.hbaseinput *initilization function*

Description

initilization function

Usage

```
r2t.hbaseinput(table, colspec = NULL, rows = NULL, caching = 1000, cacheBlocks = FALSE, autoRedu
```

Arguments

table

colspec

rows

caching

cacheBlocks
autoReduceDetect

jars

zooinfo

filter

batch

fulltable

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (table, colspec = NULL, rows = NULL, caching = 1000,
    cacheBlocks = FALSE, autoReduceDetect = FALSE, jars = "",
    zooinfo, filter = "", batch = 1L, fulltable = 0)
{
    makeRaw <- function(a) {
        J("org.apache.commons.codec.binary.Base64")$encodeBase64String(.jbyte(a))
    }
    table <- eval(table)
    colspec <- eval(colspec)
    rows <- eval(rows)
    cacheBlocks <- eval(cacheBlocks)
    autoReduceDetect <- eval(autoReduceDetect)
    caching <- eval(caching)
    function(mapred, direction, callers) {
        if (is.null(table))
            stop("Please provide table type e.g. tsdb")
        mapred$rhipe.hbase.tablename <- as.character(table[1])
        mapred$rhipe.hbase.colspec <- NULL
        if (!is.null(rows)) {
            mapred$rhipe.hbase.rowlim.start <- rows[[1]]
            mapred$rhipe.hbase.rowlim.end <- rows[[2]]
        }
        mapred$rhipe.hbase.filter <- filter
        mapred$rhipe.hbase.set.batch <- batch
        mapred$parse.ifolder = ""
        mapred$rhipe.hbase.mozilla.cacheblocks <- sprintf("%s:%s",
            as.integer(caching), as.integer(cacheBlocks))
        mapred$zookeeper.znode.parent <- zooinfo$zookeeper.znode.parent
        mapred$hbase.zookeeper.quorum <- zooinfo$hbase.zookeeper.quorum
        message(sprintf("Using %s table", table))
        mapred$rhipe.hbase.dateformat <- "yyyyMMdd"
```

```
        mapred$rhipe.hbase.mozilla.prefix <- "byteprefix"
        mapred$rhipe_inputformat_class <- "RHScanTable"
        mapred$rhipe_inputformat_keyclass <- "org.godhuli.rhipe.RHBytesWritable"
        mapred$rhipe_inputformat_valueclass <- "RHResult"
        mapred$jarfiles <- jars
        mapred
    }
  }
```

`r2t.init` *initilization function*

Description

initilization function

Arguments

`requestAdmin`

`otherConfigs`

`HBASE.HOME`

`HADOOP.HOME`

`HADOOP.CONF`

`HBASE.CONF`

`HBASE.LIB`

`rhipeJar`

`r2timeJar`

Author(s)

Bikash Agrawal

`r2t.intBittoFloat` *initilization function*

Description

initilization function

Usage

```
r2t.intBittoFloat(i)
```

Arguments

`i`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (i)
{
    sign <- bitAnd(i, 2147483648)
    if (sign == 0)
        sign <- 1
    else sign <- -1
    exp <- bitShiftR(bitAnd(i, 2139095040), 23)
    man <- bitAnd(i, 8388607)
    man <- bitOr(man, 8388608)
    f <- sign * man * (2^(exp - 150))
  }
```

`r2t.long2byte` *initilization function*

Description

initilization function

Usage

```
r2t.long2byte(ra, data)
```

Arguments

`ra`

`data`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, data)
{
    res <- .jcall(ra, "[B", "getTagK", data)
  }
```

r2t.rhwatch *initilization function*

Description

initilization function

Usage

```
r2t.rhwatch(table, sdate, edate, metrics, tagk, tagv, caching = 1400L, cacheBlocks = FALSE, autc
```

Arguments

table

sdate

edate

metrics

tagk

tagv

caching

cacheBlocks

autoReduceDetect

batch

jars

zooinfo

fulltable

output

jobname

mapred

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (table, sdate, edate, metrics, tagk, tagv, caching = 1400L,
    cacheBlocks = FALSE, autoReduceDetect = FALSE, batch = 100,
    jars = "", zooinfo, fulltable = 0, output = "", jobname = "MapReduce job",
    mapred = "")
{
    r <- r2t.getRowkeyFilter(sdate, edate, metrics, tagk, tagv)
    rows <- c(r[1], r[2])
    filter <- r[3]
    z <- rhwatch(map = map, reduce = reduce, input = r2t.hbaseinput(table = table,
        rows = rows, caching = 1400L, cacheBlocks = FALSE, jars = jars,
```

```
        zooinfo = zooinfo, filter = filter, batch = 100), output = output,
        jobname = jobname, mapred = mapred, param = list(beginningOflastMonth = Sys.Date() -
            45))
  }
```

---

r2t.setInt *initilization function*

Description

initilization function

Usage

```
r2t.setInt(ra, data, n, offset)
```

Arguments

ra

data

n

offset

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (ra, data, n, offset)
{
    .jcall(ra, "V", "setInt", data, n, offset)
  }
```

---

r2t.String2Bytes *initilization function*

Description

initilization function

Usage

```
r2t.String2Bytes(rowkey)
```

Arguments

rowkey

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (rowkey)
{
    obj <- .jnew("DataType")
    res <- .jcall(obj, "[B", "String2Bytes", rowkey)
  }
```

`r2t.toFloat` *initilization function*

Description

initilization function

Usage

```
r2t.toFloat(x)
```

Arguments

```
x
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
    if (length(x) > 7)
        return(0)
    else {
        i <- r2t.toInt(x)
        f <- r2t.intBittoFloat(i)
        return(f)
    }
  }
```

r2t.toInt *Conversion to int function*

Description

Conversion to int function

Usage

```
r2t.toInt(x)
```

Arguments

x

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
    n <- 0
    for (i in 1:length(x)) {
        n <- bitShiftL(n, 8)
        n <- bitXor(n, bitAnd(x[i], 255))
    }
    return(n)
  }
```

Index