`Rlabkey-package` *Import/export data between a labkey database and an R session*

Description

This package allows the transfer of data between a labkey database and an R session as well as the ability to modify data in the labkey database from an R seession. Data can be imported from a labkey database into R by specifying the query schema information (`labkey.selectRows`) or by using sql commands (`labkey.executeSql`). From an R session, a user can also send data to update existing data (`labkey.updateRows`) or to be inserted as new data (`labkey.insertRows`) or specify rows of data to be deleted from the labkey database (`labkey.deleteRows`).

The user must have the appropriate authorization on the labkey server in order to modify data in the database through the use of these functions.

Details

Package:	Rlabkey
Type:	Package
Version:	0.0.1
Date:	2008-08-18
License:	Apache 2.0
LazyLoad:	yes

Using this package to access a password protected labkey data base requires that the user has their login information in a.netrc file. The .netrc file contains configuration and autologin information for the File Transfer Protocol client (ftp) and other programs such as CURL. The file should be located in the users home directory and the permissions on the file should be unreadable for everybody except the owner. Permissions can be set in UNIX from the command line with chmod 600 .netrc.

On a windows machine, create and environment variable called 'HOME' that is set to your home directory ('c:/Users/<User-Name>' on Vista) or any directory you want to use. In that directory, create a text file named _netrc (note that it's underscore netrc, not dot netrc like it is on UNIX).

The following three lines must be included in the .netrc or _netrc file either separated by white space (spaces, tabs, or newlines) or commas.

machine <remote-machine-name>
login <user-email>
password <user-password>

An example would be:
machine atlas.scharp.org
login vobencha@fhcrc.org
password mypassword

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`, `labkey.executeSql`, `makeFilter`, `labkey.insertRows`, `labkey.updateRows`, `labkey.deleteRows`

`labkey.deleteRows` *Delete rows of data in a labkey database*

Description

From an R session, specify which row(s) of data should be deleted from the database.

Usage

```
labkey.deleteRows(baseUrl, folderPath, schemaName, queryName, toDelete, stripAllHidden = TRUE)
```

Arguments

`baseUrl` a string specifying the `baseUrl`for the HTTP request

`folderPath` a string specifying the `folderPath` for the HTTP request

`schemaName` a string specifying the `schemaName` for the HTTP request

`queryName` a string specifying the `queryName` for the HTTP request

`toDelete` a data frame containing a single column of data containing the data identifiers of the rows to be deleted

`stripAllHidden`
(optional) a logical value indicating whether or not to save data columns that would normally be hidden from user veiw. If no value is specified, no hidden columns are returned.

Details

A single row or multiple rows of data can be deleted. The `toDelete` data frame should contain a single column of data which are the values of the data identifiers. The colname of the data frame should be the column name of the data identifier (e.g., lsid). The data frame must be created with the `stringsAsFactors` set to FALSE.

NOTE: The column name of a column of data is the name used in the database and API. The column caption is the caption that is visable in the dataset on the web site. To correctly identify the column name, use the “export to R script” option available as a drop down option under the “views” tab for each dataset.

Value

A list is returned with name/value categories of command, rowsAffected, rows, queryName, containerPath and schemaName. The **schemaName**, **queryName** and **containerPath** properties contain the same schema, query and folder path used in the request. The **rowsAffected** property indicates he number of rows affected by the API action. This will typically be the same number as passed in the request. The **rows** property contains a list of row objects corresponding to the rows deleted.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`, `labkey.executeSql`, `makeFilter`, `labkey.insertRows`, `labkey.updateRows`

Examples

```
# Examples to be updated when labkey.org has 8.3
#mydf <- data.frame(lsid=c('urn:lsid:labkey.org:****','urn:lsid:labkey.org:****','urn:lsid:labkey.org.***

#mydata <- labkey.deleteRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo", schemaName=
```

`labkey.executeSql` *Retrieve data from a labkey database using SQL commands*

Description

Use Sql commands to specify data to be imported into R. Prior to import, data can be manipulated through standard SQL commands supported in labkey SQL.

Usage

```
labkey.executeSql(baseUrl, folderPath, schemaName, sql, maxRows = NULL,
rowOffset = NULL, stripAllHidden = TRUE)
```

Arguments

`baseUrl` a string specifying the `baseUrl`for the labkey server

`folderPath` a string specifying the `folderPath`

`schemaName` a string specifying the `schemaName` for the query

`sql` a string containing the `sql` commands to be executed

`maxRows` (optional) an integer specifying the maximum number of rows to return. If no value is specified, all rows are returned.

`rowOffset` (optional) an integer specifying which row of data should be the first row in the retrieval. If no value is specified, rows will begin at the start of the result set.

`stripAllHidden` (optional) a logical value indicating whether or not to save data columns that would normally be hidden from user veiw. If no value is specified, no hidden columns are returned.

Details

Rows returned from the SQL query can be imported into an R data frame using the `labkey.executeSql` function. The function accepts as its arguments components of the url that identify the location of the data and what SQL actions should be taken on the data prior to import. Data are returned in a data frame with column names as they appear in on the labkey database website.

Value

The requested data are returned in a data frame.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`, `makeFilter`, `labkey.insertRows`, `labkey.updateRows`, `labkey.deleteRows`

Examples

```
library(Rlabkey)

# Retrieve participant id, visit date and hemoglobin from Lab Results table
# from www.labkey.org
### NOTE: This won't work until 8.3 is up on www.labkey.org ####
```

```
#mydata <- labkey.executeSql(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo",
# schemaName="study", sql= 'select "Lab Results".ParticipantId, "Lab Results".Labdt,
# "Lab Results".Labhemo from "Lab Results"')
```

`labkey.insertRows` *Insert new rows of data into a labkey database*

Description

Send data from an R session to be inserted into the database as new rows.

Usage

```
labkey.insertRows(baseUrl, folderPath, schemaName, queryName, toInsert, stripAllHidden = TRUE)
```

Arguments

`baseUrl` a string specifying the `baseUrl`for the labkey server

`folderPath` a string specifying the `folderPath`

`schemaName` a string specifying the `schemaName` for the query

`queryName` a string specifying the `queryName`

`toInsert` a data frame containing rows of data to be inserted

`stripAllHidden` (optional) a logical value indicating whether or not to save data columns that would normally be hidden from user veiw. If no value is specified, no hidden columns are returned.

Details

A single row or multiple rows of data can be inserted. The `toInsert` data frame should contain all of the rows of data to be inserted and must be created with the `stringsAsFactors` option set to FALSE. The colname of the data frame should be the column names of the data. When inserting data into a study dataset, the sequence number must be specified.

NOTE: The column name of a column of data is the name used in the database and API. The column caption is the caption that is visable in the dataset on the web site. To correctly identify the column name, use the “export to R script” option available as a drop down option under the “views” tab for each dataset.

Value

A list is returned with name/value categories of command, rowsAffected, rows, queryName, containerPath and schemaName. The **schemaName**, **queryName** and **containerPath** properties contain the same schema, query and folder path used in the request. The **rowsAffected** property indicates he number of rows affected by the API action. This will typically be the same number as passed in the request. The **rows** property contains a list of row objects corresponding to the rows inserted.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`, `labkey.executeSql`, `makeFilter`, `labkey.updateRows`, `labkey.deleteRows`

Examples

```
# Example to be modified when 8.3 is on labkey.org
# Insert some data:
#mydf <- data.frame(SequenceNum=c(2,3), lsid=c("URG345","URG346"), participantId=c(5055, 5056), stringsAs

#mydata <- labkey.insertRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo", schemaName=
```

`labkey.selectRows` *Retrieve data from a labkey database using url specifications*

Description

Use simple filters and sorts to specify data to be imported into R. Prior to import, the data columns can be sorted, specific columns or number of rows can be requested and data filters can be applied.

Usage

```
labkey.selectRows(baseUrl, folderPath, schemaName, queryName, viewName = NULL,
colSelect = NULL, maxRows = NULL, rowOffset = NULL, colSort = NULL,
colFilter = NULL, stripAllHidden = TRUE)
```

Arguments

`baseUrl` a string specifying the `baseUrl`for the labkey server

`folderPath` a string specifying the `folderPath`

`schemaName` a string specifying the `schemaName` for the query

`queryName` a string specifying the `queryName`

`viewName` (optional) a string specifying the `viewName`

`colSelect` (optional) a vector of comma separated strings specifying which columns of a dataset or view to import `maxRows` (optional) an integer specifying how many rows of data to return. If no value is specified, all rows are returned.

`colSort` (optional) a string including the name of the column to sort preceeded by a "+" or "-" to indicate sort direction

`rowOffset` (optional) an integer specifying which row of data should be the first row in the retrieval. If no value is specified, the retrieval starts with the first row.

`colFilter` (optional) a vector or array object created by the `makeFilter` function which contains the column name, operator and value of the filter(s) to be applied to the retrieved data.

`stripAllHidden`
(optional) a logical value indicating whether or not to save data columns that would normally be hiddenfrom user view. If no value is specified, no hidden columns are returned.

Details

A full dataset or user saved view can be imported into an R data frame using the `labkey.selectRows` function. The function accepts as its arguments the components of the url that identify the location of the data and what actions should be taken on the data prior to import (ie, sorting, selecting particular columns or maximum number of rows, etc.). Data are returned in a data frame with column names as they appear on the labkey database website.

Use care when specifying column names for the colSelect or colFilter arguments. Often the column name that appears on the web site is not the same as the column caption. The column name is what is needed in colselect and colFilter arguments.

When importing data from ATLAS.scharp.org, a quick and simple way to identify the necessary components of the url (i.e., schemaName, queryName, viewName, the correct column names etc.) is to use the "export to R script" option avaiable as a drop down options under the "views" tab for each dataset.

Value

The requested data are returned in a data frame.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.executeSql`, `makeFilter`, `labkey.insertRows`, `labkey.updateRows`, `labkey.deleteRows`

Examples

```
## Retrieving data from the Labkey.org web site:

library(Rlabkey)

# Retrieve HIV Test Results and plot Western Blot data
getdata <- labkey.selectRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo",
                                schemaName="study", queryName="HIV Test Results")
plot(factor(getdata$"HIV Western Blot"), main="HIV Western Blot")

# Select columns and apply filters
myfilters<- makeFilter(c("HIVLoadQuant","GREATER_THAN",500), c("HIVRapidTest","EQUALS","Positive"))

getdata <- labkey.selectRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo",
schemaName="study", queryName="HIV Test Results",
colSelect=c("ParticipantId","HIVDate","HIVLoadQuant","HIVRapidTest"), colFilter=myfilters)
```

`labkey.updateRows` *Update existing rows of data in a labkey database*

Description

Send data from an R session to update existing rows of data in the database.

Usage

```
labkey.updateRows(baseUrl, folderPath, schemaName, queryName, toUpdate, stripAllHidden = TRUE)
```

Arguments

`baseUrl`	a string specifying the `baseUrl`for the labkey server
`folderPath`	a string specifying the `folderPath`
`schemaName`	a string specifying the `schemaName`for the query
`queryName`	a string specifying the `queryName`
`toUpdate`	a data frame containing the row(s) of data to be updated
`stripAllHidden`	(optional) a logical value indicating whether or not to save data columns that would normally be hidden from user veiw. If no value is specified, no hidden columns are returned.

Details

A single row or multiple rows of data can be updated. The `toUpdate` data frame should contain all of the rows of data to be updated and must be created with the `stringsAsFactors` option set to FALSE. The colname of the data frame should be the column names of the data.

NOTE: The column name of a column of data is the name used in the database and API. The column caption is the caption that is visable in the dataset on the web site. To correctly identify the column name, use the "export to R script" option available as a drop down option under the "views" tab for each dataset.

Value

A list is returned with name/value categories of command, rowsAffected, rows, queryName, containerPath and schemaName. The **schemaName**, **queryName** and **containerPath** properties contain the same schema, query and folder path used in the request. The **rowsAffected** property indicates he number of rows affected by the API action. This will typically be the same number as passed in the request. The **rows** property contains a list of row objects corresponding to the rows updated.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`, `labkey.executeSql`, `makeFilter`, `labkey.insertRows`, `labkey.deleteRows`

Examples

```
# Examples to be updated when labkey.org has 8.3
#mydf=data.frame(lsid=c('urn:lsid:labkey.org:****','urn:lsid:labkey.org:****') , ParticipantId=c(70001004

#mydata <- labkey.updateRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo", schemaName=
```

`makeFilter` *Builds an array of filters*

Description

This function takes inputs of column name, filter value and filter operator and returns an array of filters to be used in `labkey.selectRows`.

Usage

```
makeFilter(c(colname, operator,value))
```

Arguments

`colname`	a string specifying the name of the column to be filtered
`operator`	a string specifying what operator should be used in the filter (see options below)
`value`	an integer or string specifying the value the columns should be filtered on

Details

These filters are applied to the data prior to import into R. Currently this function allows the user to specify up to five filters. Multiple filters can be entered named or un-named vectors (see examples below).

Possible operator values are as follows: ”EQUALS”, ”EQUALS_ONE_OF”, ”NOT_EQUALS”, ”GREATER_THAN”, ”GREATER_THAN_OR_EQUAL_TO”, ”LESS_THAN”, ”LESS_THAN_OR_EQUAL_TO”, ”DATE_EQUAL”, ”DATE_NOT_EQUAL”, ”NOT_EQUAL_OR_NULL”, ”IS_NULL”, ”IS_NOT_NULL”, ”CONTAINS”, and ”DOES_NOT_CONTAIN”.

Value

The function returns either a single string or an array of strings to be use in the `colFilter` argument of the `labkey.selectRows` function.

Author(s)

Valerie Obenchain

References

http://www.omegahat.org/RCurl/,
http://dssm.unipa.it/CRAN/web/packages/rjson/rjson.pdf,
https://www.labkey.org/project/home/begin.view

See Also

`labkey.selectRows`

Examples

```
# Specify two filters:
myfilters<- makeFilter(c("HIVLoadQuant","GREATER_THAN",500), c("HIVRapidTest","EQUALS","Positive"))

# Filter using "equals one of" operator:
myfilter2 <- makeFilter(filter1=c("HIVLoadIneq","EQUALS_ONE_OF","Equals ; Less than"))

# Use in labkey.selectRows function
getdata <- labkey.selectRows(baseUrl="https://www.labkey.org", folderPath="/home/Study/demo",
schemaName="study", queryName="HIV Test Results",
colSelect=c("ParticipantId","HIVDate","HIVLoadQuant","HIVRapidTest"), colFilter=myfilters)
```