

Description

This package allows the import of data from a labkey data base into an R data frame through the use of Sql commands or by specifying the query schema information.

Data in a labkey data base can be modified from an R session by using the `insert`, `update`, and `delete` functions. The user must have the appropriate authorization on the labkey server in order to modify data in the data base 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). The file should be located in your home directory and the permissions on the file should be unreadable for everybody except the owner. Permissions can be set with the chmod command from the unix command line as `chmod 600 .netrc`. ***Insert how to do this for windows. The following three lines must be in your .netrc file: machine machinename login mylogin password mypassword

An example would be: machine atlas.scharrp.org login vobench@fhcrc.org password mypassword

See <http://linux.about.com/library/cmd/blcmdl5.netrc.htm> for more information on how to configure the .netrc file.

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](#)

<code>labkey.deleteRows</code>	<i>Delete rows of data in a labkey database</i>
--------------------------------	---

Description

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

Usage

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

Arguments

<code>baseUrl</code>	a string specifying the <code>baseUrl</code> for the HTTP request
<code>folderPath</code>	a string specifying the <code>folderPath</code> for the HTTP request
<code>schemaName</code>	a string specifying the <code>schemaName</code> for the HTTP request
<code>queryName</code>	a string specifying the <code>queryName</code> for the HTTP request
<code>toDelete</code>	a list containing the column name of the "key" or "data identifier" and the corresponding identification number of the row(s) of data to be deleted
<code>stripAllHidden</code>	[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 at a time.

Value

Information returned to the user will include the `schemaName` and the `queryName` used in the update as well as the number of rows affected (ie, 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](#)

<code>labkey.executeSql</code>	<i>Retrieve data from a labkey database using Sql commands</i>
--------------------------------	--

Description

Use Sql commands to specify data to be imported into R. Prior to import, data can be manipulated through all standard Sql commands.

Usage

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

Arguments

<code>baseUrl</code>	a string specifying the <code>baseUrl</code> for the HTTP request
<code>folderPath</code>	a string specifying the <code>folderPath</code> for the HTTP request
<code>schemaName</code>	a string specifying the <code>schemaName</code> for the HTTP request
<code>sql</code>	a string containing the <code>sql</code> commands to be executed
<code>maxRows</code>	(optional) an integer specifying how many rows of data to return. If no value is specified, all rows are returned.
<code>rowOffset</code>	(optional) an integer specifying which row of data should be the first row in the retrieval. If no value is specified, all rows are returned.
<code>stripAllHidden</code>	(optional) a logical value indicating whether or not to save data columns that would normally be hidden from 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.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](#)

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=
#                               sql= 'select "Lab Results".ParticipantId, "Lab Results".Labdt, "Lab Resul
```

<code>labkey.insertRows</code>	<i>Insert new rows of data into a labkey database</i>
--------------------------------	---

Description

Send data from an R session to a labkey server to insert new rows of data in the database.

Usage

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

Arguments

<code>baseUrl</code>	a string specifying the <code>baseUrl</code> for the HTTP request
<code>folderPath</code>	a string specifying the <code>folderPath</code> for the HTTP request
<code>schemaName</code>	a string specifying the <code>schemaName</code> for the HTTP request
<code>queryName</code>	a string specifying the <code>queryName</code> for the HTTP request
<code>toInsert</code>	a list containing field names and the corresponding data values to be inserted
<code>stripAllHidden</code>	[optional] a logical value indicating whether or not to save data columns that would normally be hidden from user view. If no value is specified, no hidden columns are returned.

Details

A single row or multiple rows of data can be inserted at a time.

Value

Information returned to the user will include the `schemaName` and the `queryName` used in the insert as well as the number of rows affected and the data sent.

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](#)

<code>labkey.selectRows</code>	<i>Retrieve data from a labkey database using url specifications</i>
--------------------------------	--

Description

Use url to specify data to be imported into R. Prior to import, 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 =  
                  colFilter = NULL, stripAllHidden = TRUE)
```

Arguments

<code>baseUrl</code>	a string specifying the <code>baseUrl</code> for the HTTP request
<code>folderPath</code>	a string specifying the <code>folderPath</code> for the HTTP request
<code>schemaName</code>	a string specifying the <code>schemaName</code> for the HTTP request
<code>queryName</code>	a string specifying the <code>queryName</code> for the HTTP request
<code>viewName</code>	(optional) a string specifying the <code>viewName</code> for the HTTP request
<code>colSelect</code>	(optional) a vector of comma separated strings specifying which columns of a dataset or view to import
<code>maxRows</code>	(optional) an integer specifying how many rows of data to return. If no value is specified, all rows are returned.
<code>colSort</code>	(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 <code>makeFilter</code> 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 hidden from 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 is not the same as the column header as seen on the web site. ***More help here*****

When importing data from ATLAS.scharp.org, a quick and simple way to identify the necessary components of the url (ie, `schemaName`, `queryName`, `viewName`, etc.) is to use the "export to R script" option available as a drop down 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](#)

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=
```

labkey.updateRows	<i>Update rows of data in a labkey database</i>
-------------------	---

Description

Send data from an R session to a labkey server to update 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 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
toUpdate	a list containing the name of the field and the corresponding data values to be updated
stripAllHidden	[optional] a logical value indicating whether or not to save data columns that would normally be hidden from user view. If no value is specified, no hidden columns are returned.

Details

A single row or multiple rows of data can be updated at a time.

Value

Information returned to the user will include the **schemaName** and the **queryName** used in the update as well as the number of rows affected and the data sent in the update.

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](#)

<code>makeFilter</code>	<i>Builds an array of filters</i>
-------------------------	-----------------------------------

Description

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

Usage

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

Arguments

<code>colname</code>	a string specifying the name of the column to be filtered
<code>operator</code>	a text string specifying what operator should be used in the filter
<code>value</code>	an integer or string specifying the value the columns should be filtered on

Details

Possible operator values are as follows: "EQUALS", "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".

Multiple filters can be applied (see examples). Currently this function supports specifying up to five filters.

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

```
# Specification of 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=
```