

Package ‘rrefine’

October 7, 2020

Type Package

Title r Client for OpenRefine API

Version 1.1.0

Date 2020-10-04

Maintainer VP Nagraj <nagraj@nagraj.net>

Description 'OpenRefine' (formerly 'Google Refine') is a popular, open source data cleaning software. This package enables users to programmatically trigger data transfer between R and 'OpenRefine'. Available functionality includes project import, export and deletion.

License GPL-3

LazyData TRUE

RoxygenNote 7.1.1

Imports httr (>= 1.1.0), rlist, readr

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL <https://github.com/vpnagraj/rrefine>

BugReports <https://github.com/vpnagraj/rrefine/issues>

Encoding UTF-8

NeedsCompilation no

Author VP Nagraj [aut, cre]

Repository CRAN

Date/Publication 2020-10-07 15:20:06 UTC

R topics documented:

lateformeeting	2
lfm_clean	2
refine_check	3
refine_delete	4
refine_export	5

refine_id	6
refine_metadata	6
refine_path	7
refine_query	8
refine_token	8
refine_upload	9

Index 10

lateformeeting	<i>a "dirty" data set to demonstrate rrefine features</i>
----------------	---

Description

This data is a simulated collection of dates, days of the week, numbers of hours slept and indicators of whether or not the subject was on time for work. All observations appearing in this data set are fictitious, and any resemblance to actual arrival times for work is purely coincidental.

Usage

```
lateformeeting
```

Format

A data frame with 63 rows and 4 variables

- theDate date of observation in varying formats
- what.day.whas.it day of the week in varying formats
- sleephours number of hours slept
- was.i.on.time.for.work indicator of on-time arrival to work

Examples

```
head(lateformeeting)
```

lfm_clean	<i>a "clean" version of the lateformeeting sample data set</i>
-----------	--

Description

This data is a simulated collection of dates, days of the week, numbers of hours slept and indicators of whether or not the subject was on time for work. All observations appearing in this data set are fictitious, and any resemblance to actual arrival times for work is purely coincidental.

Usage

```
lfm_clean
```

Format

A data frame with 63 rows and 4 variables

- date date of observation in POSIXct format
- dotw day of the week in consistent format
- hours.slept number of hours slept
- on.time indicator of on-time arrival to work

Examples

```
head(lfm_clean)
```

refine_check

Helper function to check if rrefine can connect to OpenRefine

Description

This function will check that rrefine is able to access the running OpenRefine instance. Used internally prior to upload, delete, and export operations.

Usage

```
refine_check(...)
```

Arguments

... Additional parameters to be inherited by [refine_path](#); allows users to specify host and port arguments if the OpenRefine instance is running at a location other than `http://127.0.0.1:3333`

Value

Error message if rrefine is unable to connect to OpenRefine, otherwise is invisible

refine_delete	<i>Delete project from OpenRefine</i>
---------------	---------------------------------------

Description

This function allows users to delete a project in OpenRefine by name or unique project identifier. By default users are prompted to confirm deletion. The function wraps the OpenRefine API `/command/core/delete-project` query.

Usage

```
refine_delete(project.name = NULL, project.id = NULL, force = FALSE, ...)
```

Arguments

<code>project.name</code>	Name of project to be deleted
<code>project.id</code>	Unique identifier for open refine project to be deleted
<code>force</code>	Boolean indicating whether or not the prompt to confirm deletion should be skipped; default is FALSE
<code>...</code>	Additional parameters to be inherited by refine_path ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>

Value

Operates as a side-effect to delete the project. Issues a message that the project has been deleted.

References

<https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API#delete-project>

Examples

```
## Not run:  
fp <- system.file("extdata", "lateformeeting.csv", package = "rrefine")  
refine_upload(fp, project.name = "lfm")  
refine_delete("lfm", force = TRUE)  
  
## End(Not run)
```

refine_export	<i>Export data from OpenRefine</i>
---------------	------------------------------------

Description

This function allows users to pull data from a running OpenRefine instance into R. Users can specify project by name or unique identifier. The function wraps the OpenRefine API query to `/command/core/export-rows` and currently only supports export of data in tabular format.

Usage

```
refine_export(  
  project.name = NULL,  
  project.id = NULL,  
  format = "csv",  
  col.names = TRUE,  
  encoding = "UTF-8",  
  ...  
)
```

Arguments

<code>project.name</code>	Name of project to be exported
<code>project.id</code>	Unique identifier for project to be exported
<code>format</code>	File format of project to be exported; note that the only current supported options are 'csv' or 'tsv'
<code>col.names</code>	Logical indicator for whether column names should be included; default is TRUE
<code>encoding</code>	Character encoding for exported data; default is UTF-8
<code>...</code>	Additional parameters to be inherited by refine_path ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>

Value

A tibble that has been parsed and read into memory using [read_csv](#). If `col.names=TRUE` then the tibble will have column headers.

References

<https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API#export-rows>

Examples

```
## Not run:
fp <- system.file("extdata", "lateformeeting.csv", package = "rrefine")
refine_upload(fp, project.name = "lfm")
refine_export("lfm", format = "csv")

## End(Not run)
```

refine_id	<i>Helper function to get OpenRefine project.id by project.name</i>
-----------	---

Description

For functions that allow either a project name or id to be passed, this function is used internally to resolve the project id from name if necessary. It also validates that values passed to the ‘project.id’ argument match an existing project id in the running OpenRefine instance.

Usage

```
refine_id(project.name, project.id, ...)
```

Arguments

project.name	Name of project
project.id	Unique identifier for project
...	Additional parameters to be inherited by refine_path ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>

Value

Unique id of project

refine_metadata	<i>Get all project metadata from OpenRefine</i>
-----------------	---

Description

This function is included internally to help retrieve metadata from the running OpenRefine instance. The query uses the OpenRefine API `/command/core/get-all-project-metadata` endpoint.

Usage

```
refine_metadata(...)
```

Arguments

... Additional parameters to be inherited by `refine_path`; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than `http://127.0.0.1:3333`

Value

Parsed list object with all project metadata including identifiers, names, dates of creation and modification, tags and more.

References

<https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API#get-all-projects-metadata>

Examples

```
## Not run:  
refine_metadata()  
  
## End(Not run)
```

refine_path

Helper function to configure and call path to OpenRefine

Description

This function is a helper that is used throughout `rrefine` to construct the path to the OpenRefine instance. By default this points to the localhost (`http://127.0.0.1:3333`).

Usage

```
refine_path(host = "http://127.0.0.1", port = "3333")
```

Arguments

host Host for running OpenRefine instance; default is `http://127.0.0.1`
port Port number for running OpenRefine instance; default is `3333`

Value

Character vector with path to running OpenRefine instance

refine_query	<i>Helper function to build OpenRefine API query</i>
--------------	--

Description

Starting with the path to the running instance, this function will add a query command and (optionally) a CSFR token with [refine_token](#)

Usage

```
refine_query(query, use_token = TRUE, ...)
```

Arguments

query	Character vector specifying the API endpoint to query
use_token	Boolean indicating whether or not the query string should include a CSRF Token (see refine_token ; default is TRUE)
...	Additional parameters to be inherited by refine_path ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>

Value

Character vector with query based on parameter entered

refine_token	<i>Helper function to retrieve CSFR token</i>
--------------	---

Description

Helper function to retrieve CSFR token

Usage

```
refine_token(...)
```

Arguments

...	Additional parameters to be inherited by refine_path ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>
-----	--

Value

Character vector with OpenRefine CSFR token

refine_upload	<i>Upload a file to OpenRefine</i>
---------------	------------------------------------

Description

This function attempts to upload contents of a file and create a new project in OpenRefine. Users can optionally navigate directly to the running instance to interact with the project. The function wraps the OpenRefine API `/command/core/create-project-from-upload` query.

Usage

```
refine_upload(file, project.name = NULL, open.browser = FALSE, ...)
```

Arguments

<code>file</code>	Path to file to upload; upload format is inferred from the file extension, and currently only ".csv" and ".tsv" files are allowed.
<code>project.name</code>	Optional parameter to specify name of the project to be created upon upload; default is NULL and project will be named 'Untitled' in OpenRefine
<code>open.browser</code>	Boolean for whether or not the browser should open on successful upload; default is FALSE
<code>...</code>	Additional parameters to be inherited by <code>refine_path</code> ; allows users to specify host and port arguments if the OpenRefine instance is running at a location other than <code>http://127.0.0.1:3333</code>

Value

Operates as a side-effect, either opening a browser and pointing to the OpenRefine instance (if `open.browser=TRUE`) or issuing a message.

References

<https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API#create-project>

Examples

```
## Not run:
fp <- system.file("extdata", "lateformeeting.csv", package = "rrefine")
refine_upload(fp, project.name = "lfm")
write.table(x = mtcars, file = "mtcars.tsv", sep = "\t")
refine_upload(file = "mtcars.tsv", project.name = "mtcars")

## End(Not run)
```

Index

* datasets

lateformeeting, [2](#)

lfm_clean, [2](#)

lateformeeting, [2](#)

lfm_clean, [2](#)

read_csv, [5](#)

refine_check, [3](#)

refine_delete, [4](#)

refine_export, [5](#)

refine_id, [6](#)

refine_metadata, [6](#)

refine_path, [3-7](#), [7](#), [8](#), [9](#)

refine_query, [8](#)

refine_token, [8](#), [8](#)

refine_upload, [9](#)