

Package ‘qualtRics’

April 12, 2019

Type Package

Title Download 'Qualtrics' Survey Data

Version 3.1.0

Description Provides functions to access survey results directly into R using the 'Qualtrics' API. 'Qualtrics' <<https://www.qualtrics.com/about/>> is an online survey and data collection software platform. See <<https://api.qualtrics.com/>> for more information about the 'Qualtrics' API. This package is community-maintained and is not officially supported by 'Qualtrics'.

URL <https://ropensci.github.io/qualtRics/>,
<https://github.com/ropensci/qualtRics>

BugReports <https://github.com/ropensci/qualtRics/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

Imports httr, stringr, readr, jsonlite, assertthat, sjlabelled, yaml,
dplyr, rlang

Suggests knitr, rmarkdown, testthat, httpptest

VignetteBuilder knitr

NeedsCompilation no

Author Jasper Ginn [aut],
Samuel Kaminsky [ctb],
Eric Knudsen [ctb],
Julia Silge [aut, cre] (<<https://orcid.org/0000-0002-3671-836X>>),
Phoebe Wong [ctb]

Maintainer Julia Silge <julia.silge@gmail.com>

Repository CRAN

Date/Publication 2019-04-12 10:13:49 UTC

R topics documented:

all_surveys	2
append_root_url	3
check_for_warnings	4
check_params	4
construct_header	4
create_raw_payload	5
download_qualtrics_export	6
fetch_survey	6
getSurvey	8
getSurveyQuestions	8
getSurveys	9
infer_data_types	9
metadata	9
qualtRicsConfigFile	11
qualtrics_api_credentials	12
qualtrics_api_request	13
qualtrics_response_codes	13
readSurvey	14
read_survey	14
registerOptions	15
survey_questions	17
wrapper_mc	18
Index	19

all_surveys	<i>Retrieve a data frame of all active surveys on Qualtrics</i>
-------------	---

Description

Retrieve a data frame of all active surveys on Qualtrics

Usage

```
all_surveys()
```

See Also

See <https://api.qualtrics.com/docs> for documentation on the Qualtrics API.

Examples

```
## Not run:
# Register your Qualtrics credentials if you haven't already
qualtrics_api_credentials(
  api_key = "<YOUR-API-KEY>",
  base_url = "<YOUR-BASE-URL>"
)

# Retrieve a list of all surveys
surveys <- all_surveys()

# Retrieve a single survey
mysurvey <- fetch_survey(surveyID = surveys$id[6])

mysurvey <- fetch_survey(
  surveyID = surveys$id[6],
  save_dir = tempdir(),
  start_date = "2018-01-01",
  end_date = "2018-01-31",
  limit = 100,
  label = TRUE,
  unanswer_recode = "UNANS",
  verbose = TRUE
)

## End(Not run)
```

`append_root_url`*Append proper end points to create root URL*

Description

Append proper end points to create root URL

Usage

```
append_root_url(base_url, type = c("responseexports", "surveys"))
```

Arguments

<code>base_url</code>	Base URL for your institution (see https://api.qualtrics.com/docs/root-url)
<code>type</code>	Either "responseexports" or "surveys"

Value

Root URL, plus appended end point

check_for_warnings *Check if http GET result contains a warning*

Description

Check if http GET result contains a warning

Usage

```
check_for_warnings(resp)
```

Arguments

resp object returned by [qualtrics_response_codes](#)

check_params *Check if parameters passed to functions are correct*

Description

Check if parameters passed to functions are correct

Usage

```
check_params(...)
```

Arguments

... options passed to function

construct_header *Construct a header to send to Qualtrics API*

Description

Construct a header to send to Qualtrics API

Usage

```
construct_header(API_TOKEN)
```

Arguments

API_TOKEN API token. Available in your Qualtrics account (see: <https://api.qualtrics.com/docs/authentication>)

See Also

See <https://api.qualtrics.com/docs/root-url> for documentation on the Qualtrics API.

create_raw_payload *Create raw JSON payload to post response exports request*

Description

Create raw JSON payload to post response exports request

Usage

```
create_raw_payload(surveyID, label = TRUE, last_response = NULL,  
  start_date = NULL, end_date = NULL, limit = NULL,  
  local_time = FALSE, unanswer_recode = NULL,  
  include_questions = NULL)
```

Arguments

surveyID	Survey ID
label	Flag
last_response	Flag
start_date	Flag
end_date	Flag
limit	Flag
local_time	Flag
unanswer_recode	Flag
include_questions	Flag

Value

JSON file with options to send to API

See Also

Functions [all_surveys](#) and [registerOptions](#) have more details on these parameters

download_qualtrics_export

Download response export

Description

Download response export

Usage

```
download_qualtrics_export(check_url, verbose = FALSE)
```

Arguments

check_url	URL provided by Qualtrics API that shows the download percentage completeness
verbose	See fetch_survey

fetch_survey

Download a survey and import it into R

Description

Download a Qualtrics survey you own via API and import the survey directly into R.

Usage

```
fetch_survey(surveyID, last_response = NULL, start_date = NULL,
  end_date = NULL, unanswer_recode = NULL, limit = NULL,
  include_questions = NULL, save_dir = NULL, force_request = FALSE,
  verbose = TRUE, label = TRUE, convert = TRUE, local_time = FALSE,
  ...)
```

Arguments

surveyID	String. Unique ID for the survey you want to download. Returned as id by the all_surveys function.
last_response	String. Export all responses received after the specified response ID. Defaults to NULL.
start_date	String. Filter to only exports responses recorded after the specified date. Accepts dates as character strings in format "YYYY-MM-DD". Defaults to NULL.
end_date	String. Filter to only exports responses recorded before the specified date. Accepts dates as character strings in format "YYYY-MM-DD". Defaults to NULL.

unanswer_recode	String. Recode seen but unanswered questions with a string value. Defaults to NULL.
limit	Integer. Maximum number of responses exported. Defaults to NULL (all responses).
include_questions	Vector of strings (e.g. c('QID1', 'QID2', 'QID3')). Export only specified questions. Defaults to NULL.
save_dir	String. Directory where survey results will be stored. Defaults to a temporary directory which is cleaned when your R session is terminated. This argument is useful if you'd like to store survey results. The downloaded survey will be stored as an RDS file (see readRDS).
force_request	Logical. <code>fetch_survey()</code> saves each survey in a temporary directory so that it can quickly be retrieved later. If <code>force_request</code> is TRUE, <code>fetch_survey()</code> always downloads the survey from the API instead of loading it from the temporary directory. Defaults to FALSE.
verbose	Logical. If TRUE, verbose messages will be printed to the R console. Defaults to TRUE.
label	Logical. TRUE to export survey responses as Choice Text or FALSE to export survey responses as values.
convert	Logical. If TRUE, then the <code>fetch_survey</code> function will convert certain question types (e.g. multiple choice) to proper data type in R. Defaults to TRUE.
local_time	Logical. Use local timezone to determine response date values? Defaults to FALSE. See https://api.qualtrics.com/docs/dates-and-times for more information.
...	optional arguments. You can pass all arguments listed in registerOptions (except a different base url / api key). You can also pass a argument 'fileEncoding' (see 'fileEncoding' argument in read_survey) to import your survey using a specific encoding.

See Also

See <https://api.qualtrics.com/docs/response-exports> for documentation on the Qualtrics API.

Examples

```
## Not run:
# Register your Qualtrics credentials if you haven't already
qualtrics_api_credentials(
  api_key = "<YOUR-API-KEY>",
  base_url = "<YOUR-BASE-URL>"
)

# Retrieve a list of surveys
surveys <- all_surveys()
```

```
# Retrieve a single survey
mysurvey <- fetch_survey(surveyID = surveys$id[6])

mysurvey <- fetch_survey(
  surveyID = surveys$id[6],
  save_dir = tempdir(),
  start_date = "2018-01-01",
  end_date = "2018-01-31",
  limit = 100,
  label = TRUE,
  unanswer_recode = "UNANS",
  verbose = TRUE
)

## End(Not run)
```

getSurvey	<i>Download a survey and import it into R</i>
-----------	---

Description

This function is soft deprecated; use [fetch_survey](#) instead.

Usage

```
getSurvey(...)
```

Arguments

... All arguments for `fetch_survey`

getSurveyQuestions	<i>Retrieve a data frame containing question IDs and labels</i>
--------------------	---

Description

This function is soft deprecated; use [survey_questions](#) instead.

Usage

```
getSurveyQuestions(surveyID)
```

Arguments

surveyID A string. Unique ID for the survey you want to download. Returned as 'id' by the [getSurveys](#) function.

getSurveys	<i>Retrieve a data frame of all active surveys on Qualtrics</i>
------------	---

Description

This function is soft deprecated; use [all_surveys](#) instead.

Usage

```
getSurveys()
```

infer_data_types	<i>Set proper data types on survey data.</i>
------------------	--

Description

Set proper data types on survey data.

Usage

```
infer_data_types(data, surveyID, verbose = FALSE)
```

Arguments

data	Imported Qualtrics survey
surveyID	ID of survey
verbose	Flag

metadata	<i>Download metadata for a survey</i>
----------	---------------------------------------

Description

Using this function, you can retrieve metadata about your survey. This information includes question metadata (type, options, choices, etc), number of responses, general metadata, survey flow, etc.

Usage

```
metadata(surveyID, get = list(), ...)
```

Arguments

surveyID	A string. Unique ID for the survey you want to download. Returned as 'id' by the <code>getSurveys</code> function.
get	A list containing 'TRUE'/'FALSE' values of one of the following: 'metadata', 'questions', 'responsecounts', 'blocks', 'flow', 'embedded_data', or 'comments'. A 'TRUE' value will return that specific element. If you leave this empty, the function will return the 'metadata', 'questions', and 'responsecounts' elements. See examples below for more information.
...	Additional options. User may pass an argument called 'questions', which should be a vector containing the names of questions for which you want to return metadata.

Examples

```
## Not run:
# Register your Qualtrics credentials if you haven't already
qualtrics_api_credentials(
  api_key = "<YOUR-API-KEY>",
  base_url = "<YOUR-BASE-URL>"
)

# Get an overview of surveys
surveys <- getSurveys()

# Get metadata for a survey
md <- metadata(surveyID = surveys$id[6])

# Get metadata with specific elements
md_specific <- metadata(
  surveyID = id,
  get = list(questions = FALSE, flow = TRUE)
)

# Get specific question metadata
question_specific <- metadata(
  surveyID = id,
  get = list(questions = TRUE),
  questions = c("Q1", "Q2")
)

# Example of a metadata file
file <- system.file("extdata", "metadata.rds", package = "qualtRics")

# Load
metadata_ex <- readRDS(file = file)

## End(Not run)
```

qualtRicsConfigFile *Prints an Example of a Qualtrics Configuration File to the Console.*

Description

This function is soft deprecated; use [qualtrics_api_credentials](#) instead.

Usage

```
qualtRicsConfigFile(api_token = NULL, base_url = NULL,  
  verbose = TRUE, useLabels = TRUE, convertVariables = TRUE,  
  useLocalTime = FALSE, dateWarning = TRUE, root_url = NULL)
```

Arguments

api_token	String. API token. Available in your qualtrics account (see: https://api.qualtrics.com/docs/authentication)
base_url	String. Base url for your institution (see: https://api.qualtrics.com/docs/root-url)
verbose	Logical. If TRUE, verbose messages will be printed to the R console. Defaults to TRUE.
useLabels	Logical. TRUE to export survey responses as Choice Text or FALSE to export survey responses as values.
convertVariables	Logical. If TRUE, then the getSurvey function will convert certain question types (e.g. multiple choice) to proper data type in R. Defaults to TRUE.
useLocalTime	Logical. Use local timezone to determine response date values? Defaults to FALSE. See https://api.qualtrics.com/docs/dates-and-times for more information.
dateWarning	Logical. Once per session, qualtrics will emit a warning about date conversion for surveys. You can turn this warning off by changing the flag to FALSE. Defaults to TRUE.
root_url	String. Deprecated. Use 'base_url' instead. This will be removed in future versions.

See Also

See <https://api.qualtrics.com/docs/root-url> for documentation on the Qualtrics API. See <https://github.com/ropensci/qualtrics/blob/master/README.md#using-a-configuration-file> for more information about the qualtrics configuration file.

Examples

```
## Not run:
# Execute this line to get instructions on how to make a .qualtrics.yml config file.
qualtricsConfigFile()

## End(Not run)
```

```
qualtrics_api_credentials
```

Install Qualtrics credentials in your .Renvirom file for repeated use

Description

This function adds your Qualtrics API key and base URL to your .Renvirom file so it can be called securely without being stored in your code. After you have installed these two credentials, they can be called any time with `Sys.getenv("QUALTRICS_API_KEY")` or `Sys.getenv("QUALTRICS_BASE_URL")`. If you do not have an .Renvirom file, the function will create one for you. If you already have an .Renvirom file, the function will append the key to your existing file, while making a backup of your original file for disaster recovery purposes.

Usage

```
qualtrics_api_credentials(api_key, base_url, overwrite = FALSE,
  install = FALSE)
```

Arguments

<code>api_key</code>	The API key provided to you from Qualtrics formatted in quotes. Learn more about Qualtrics API keys at https://api.qualtrics.com/docs/finding-qualtrics-ids
<code>base_url</code>	The institution-specific base URL for your Qualtrics account, formatted in quotes. Find your base URL at https://api.qualtrics.com/docs/root-url
<code>overwrite</code>	If TRUE, will overwrite existing Qualtrics credentials that you already have in your .Renvirom file.
<code>install</code>	If TRUE, will install the key in your .Renvirom file for use in future sessions. Defaults to FALSE (single session use).

Examples

```
## Not run:
qualtrics_api_credentials(
  api_key = "<YOUR-QUALTRICS_API_KEY>",
  base_url = "<YOUR-QUALTRICS_BASE_URL>",
  install = TRUE
)
# Reload your environment so you can use the credentials without restarting R
```

```

readRenviro("~/Renviro")
# You can check it with:
Sys.getenv("QUALTRICS_API_KEY")

# If you need to overwrite existing credentials:
qualtrics_api_credentials(
  api_key = "<YOUR-QUALTRICS_API_KEY>",
  base_url = "<YOUR-QUALTRICS_BASE_URL>",
  overwrite = TRUE,
  install = TRUE
)
# Reload your environment to use the credentials

## End(Not run)

```

qualtrics_api_request *Send http requests to Qualtrics API*

Description

Send http requests to Qualtrics API

Usage

```

qualtrics_api_request(verb = c("GET", "POST"), url = url,
  body = NULL)

```

Arguments

verb	Type of request to be sent (@seealso VERB)
url	Qualtrics endpoint URL created by append_root_url function
body	Options created by create_raw_payload function

qualtrics_response_codes

Checks responses against Qualtrics response codes and returns error message.

Description

Checks responses against Qualtrics response codes and returns error message.

Usage

```

qualtrics_response_codes(res, raw = FALSE)

```

Arguments

res	Response from httr::GET
raw	If TRUE, add 'raw' flag to httr::content() function.

readSurvey	<i>Read a CSV file exported from Qualtrics</i>
------------	--

Description

Reads comma separated CSV files generated by Qualtrics software. The second line containing the variable labels is imported. Repetitive introductions to matrix questions are automatically removed. Variable labels are stored as attributes.

Usage

```
readSurvey(...)
```

Arguments

...	All arguments for read_survey
-----	---

read_survey	<i>Read a CSV file exported from Qualtrics</i>
-------------	--

Description

Reads comma separated CSV files generated by Qualtrics software. The second line containing the variable labels is imported. Repetitive introductions to matrix questions are automatically removed. Variable labels are stored as attributes.

Usage

```
read_survey(file_name, strip_html = TRUE, legacy = FALSE)
```

Arguments

file_name	String. A CSV data file.
strip_html	Logical. If TRUE, then remove HTML tags. Defaults to TRUE.
legacy	Logical. If TRUE, then import "legacy" format CSV files (as of 2017). Defaults to FALSE.

Value

A data frame. Variable labels are stored as attributes. They are not printed on the console but are visible in the RStudio viewer.

Examples

```
## Not run:
# Generic use of read_survey()
df <- read_survey("<YOUR-PATH-TO-CSV-FILE>")

## End(Not run)
# Example using current data format
file <- system.file("extdata", "sample.csv", package = "qualtRics")
df <- read_survey(file)

# Example using legacy data format
file <- system.file("extdata", "sample_legacy.csv", package = "qualtRics")
df <- read_survey(file, legacy = TRUE)
```

registerOptions

Register Qualtrics API Key, Base Url and Other Options

Description

This function is soft deprecated; use [qualtrics_api_credentials](#) instead. This function registers the user's qualtrics API key, base url and other options for the remainder of the R session. This function only needs to be called once (at the beginning of each R session). You may also use a configuration file. See [qualtRicsConfigFile](#). Note that you must pass both an api token and a base url if you call this function for the first time in a session and you're not using a config file. Thereafter, you can pass these options individually.

Usage

```
registerOptions(verbose = TRUE, useLabels = TRUE,
  convertVariables = TRUE, useLocalTime = FALSE, dateWarning = TRUE,
  ...)
```

Arguments

verbose	Logical. If TRUE, verbose messages will be printed to the R console. Defaults to TRUE.
useLabels	Logical. TRUE to export survey responses as Choice Text or FALSE to export survey responses as values.
convertVariables	Logical. If TRUE, then the getSurvey function will convert certain question types (e.g. multiple choice) to proper data type in R. Defaults to TRUE.
useLocalTime	Logical. Use local timezone to determine response date values? Defaults to FALSE. See https://api.qualtrics.com/docs/dates-and-times for more information.

dateWarning Logical. Once per session, qualTRics will emit a warning about date conversion for surveys. You can turn this warning off by changing the flag to FALSE. Defaults to TRUE.

... Either one or both of 'api_token' and 'base_url' to register the Qualtrics api key and institution-specific root url manually. (see example). See also [qualTRicsConfigFile](#) for an explanation of the root_url and api_token parameters.

See Also

See <https://github.com/ropensci/qualTRics/blob/master/README.md#using-a-configuration-file> for more information about the qualTRics configuration file. See: <https://api.qualtrics.com/docs/authentication> to find your Qualtrics API key and <https://api.qualtrics.com/docs/root-url> for more information about the institution-specific root url.

Examples

```
## Not run:
# Register your Qualtrics credentials if you haven't already
# Note that you need to pass both the 'api_token' and 'base_url'
# parameters if you call this function for the first time.
registerOptions(
  api_token = "<YOUR-API-TOKEN>",
  base_url = "<YOUR-ROOT-URL>"
)
# Register a different root url
registerOptions(base_url = "<YOUR-OTHER-ROOT-URL>")
# Retrieve a list of surveys
surveys <- getSurveys()
# Retrieve a single survey
mysurvey <- getSurvey(
  surveyID = surveys$id[6],
  save_dir = tempdir(),
  verbose = TRUE
)
# You can use the same parameters as those found in the qualtrics API documentation
# Found here: https://api.qualtrics.com/docs/csv
mysurvey <- getSurvey(
  surveyID = surveys$id[6],
  save_dir = tempdir(),
  startDate = "2017-01-01",
  endDate = "2017-01-31",
  limit = 100,
  seenUnansweredRecode = "UNANS"
)

## End(Not run)
```

survey_questions	<i>Retrieve a data frame containing question IDs and labels</i>
------------------	---

Description

Retrieve a data frame containing question IDs and labels

Usage

```
survey_questions(surveyID)
```

Arguments

surveyID A string. Unique ID for the survey you want to download. Returned as 'id' by the [getSurveys](#) function.

See Also

See <https://api.qualtrics.com/docs> for documentation on the Qualtrics API.

Examples

```
## Not run:
# Register your Qualtrics credentials if you haven't already
qualtrics_api_credentials(
  api_key = "<YOUR-API-KEY>",
  base_url = "<YOUR-BASE-URL>"
)

# Retrieve a list of surveys
surveys <- all_surveys()

# Retrieve questions for a survey
questions <- survey_questions(surveyID = surveys$id[6])

# Retrieve a single survey, filtering for specific questions
mysurvey <- fetch_survey(
  surveyID = surveys$id[6],
  save_dir = tempdir(),
  include_questions = c("QID1", "QID2", "QID3"),
  verbose = TRUE
)

## End(Not run)
```

`wrapper_mc`*Convert multiple choice questions to ordered factors*

Description

Convert multiple choice questions to ordered factors

Usage

```
wrapper_mc(data, col_name, survey_meta)
```

Arguments

<code>data</code>	Imported Qualtrics survey
<code>col_name</code>	Column name
<code>survey_meta</code>	Survey metadata

Index

`all_surveys`, [2](#), [5](#), [6](#), [9](#)
`append_root_url`, [3](#), [13](#)

`check_for_warnings`, [4](#)
`check_params`, [4](#)
`construct_header`, [4](#)
`create_raw_payload`, [5](#), [13](#)

`download_qualtrics_export`, [6](#)

`fetch_survey`, [6](#), [6](#), [7](#), [8](#)

`getSurvey`, [8](#), [11](#), [15](#)
`getSurveyQuestions`, [8](#)
`getSurveys`, [8](#), [9](#), [10](#), [17](#)

`infer_data_types`, [9](#)

`metadata`, [9](#)

`qualtrics_api_credentials`, [11](#), [12](#), [15](#)
`qualtrics_api_request`, [13](#)
`qualtrics_response_codes`, [4](#), [13](#)
`qualtricsConfigFile`, [11](#), [15](#), [16](#)

`read_survey`, [7](#), [14](#), [14](#)
`readRDS`, [7](#)
`readSurvey`, [14](#)
`registerOptions`, [5](#), [7](#), [15](#)

`survey_questions`, [8](#), [17](#)

`VERB`, [13](#)

`wrapper_mc`, [18](#)