

Package ‘qualtRics’

September 14, 2021

Type Package

Title Download 'Qualtrics' Survey Data

Version 3.1.5

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

License MIT + file LICENSE

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

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

Imports assertthat, dplyr (>= 1.0), glue, httr, jsonlite, lifecycle, lubridate, purrr, readr, rlang, sjlabelled, stringr, tibble, tidyr

Suggests knitr, rmarkdown, testthat, vcr, webmockr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Jasper Ginn [aut],
Jackson Curtis [ctb],
Shaun Jackson [ctb],
Samuel Kaminsky [ctb],
Eric Knudsen [ctb],
Joseph O'Brien [ctb],
Daniel Seneca [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 2021-09-14 19:10:02 UTC

R topics documented:

all_mailinglists	2
all_surveys	3
check_for_warnings	4
check_params	4
column_map	5
construct_header	6
create_raw_payload	6
download_qualtrics_export	7
extract_colmap	8
fetch_description	8
fetch_distributions	9
fetch_id	10
fetch_mailinglist	11
fetch_survey	12
getSurvey	14
getSurveyQuestions	15
getSurveys	15
infer_data_types	15
metadata	16
qualtRicsConfigFile	17
qualtrics_api_credentials	17
qualtrics_response_codes	19
readSurvey	19
read_survey	20
survey_questions	21
wrapper_mc	22
Index	23

all_mailinglists	<i>Retrieve a data frame of all mailing lists from Qualtrics</i>
------------------	--

Description

Retrieve a data frame of all mailing lists from Qualtrics

Usage

```
all_mailinglists()
```

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 mailing lists
mailinglists <- all_mailinglists()

## End(Not run)
```

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/> 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)
```

check_for_warnings *Check if htrr GET result contains a warning*

Description

Check if htrr 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

`column_map`*Retrieve a data frame containing survey column mapping*

Description

Retrieve a data frame containing survey column mapping

Usage

```
column_map(surveyID)
```

Arguments

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

See Also

See <https://api.qualtrics.com/> 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 column mapping for a survey
mapping <- column_map(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)
```

construct_header	<i>Construct a header to send to Qualtrics API</i>
------------------	--

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/)
-----------	--

create_raw_payload	<i>Create raw JSON payload to post response exports request</i>
--------------------	---

Description

Create raw JSON payload to post response exports request

Usage

```
create_raw_payload(  
  label = TRUE,  
  start_date = NULL,  
  end_date = NULL,  
  limit = NULL,  
  time_zone = NULL,  
  unanswer_recode = NULL,  
  unanswer_recode_multi = NULL,  
  include_display_order = TRUE,  
  include_questions = NULL,  
  breakout_sets = NULL  
)
```

Arguments

label	Flag
start_date	Flag
end_date	Flag
limit	Flag

time_zone	Flag
unanswer_recode	Flag
unanswer_recode_multi	Flag
include_display_order	Flag
include_questions	Flag
breakout_sets	Flag

Value

JSON file with options to send to API

See Also

See [all_surveys](#) for more details on these parameters

download_qualtrics_export
Download response export

Description

Download response export

Usage

```
download_qualtrics_export(fetch_url, requestID, verbose = FALSE)
```

Arguments

fetch_url	URL provided by Qualtrics API that shows the download percentage completeness
requestID	ID
verbose	See fetch_survey

extract_colmap	<i>Extract column map from survey data download</i>
----------------	---

Description

Helper function to extract the column map attached to a response data download obtained from [fetch_survey](#) (using the default `add_column_map = TRUE`)

Usage

```
extract_colmap(respdata)
```

Arguments

respdata	Response data including a column map dataframe as an attribute
----------	--

Examples

```
## Not run:  
# Retrieve a list of surveys  
surveys <- all_surveys()  
  
# Retrieve a single survey  
mysurvey <- fetch_survey(surveyID = surveys$id[6])  
  
# Extract column mapping for survey  
extract_colmap(mysurvey)  
  
## End(Not run)
```

fetch_description	<i>Download complete survey description using the Qualtrics v3 "Get Survey" API endpoint.</i>
-------------------	---

Description

Download complete survey description using the Qualtrics v3 "Get Survey" API endpoint.

Usage

```
fetch_description(surveyID, elements = NULL, legacy = FALSE, ...)
```


Arguments

surveyID	A string. Unique ID for the survey you want to download. Returned as "id" by the <code>all_surveys</code> function.
elements	A character vector. Lists elements of survey definition to be maintained. Possible elements are "metadata", "surveyoptions", "flow", "blocks", "questions", "responsesets", and/or "scoring" (case-insensitive). If <code>legacy = TRUE</code> , then possible elements are "metadata", "questions", "responsecounts", "blocks", "flow", "embedded_data", and/or "comments".
legacy	Logical. If TRUE, will use older Get Survey API endpoint via a call to legacy function <code>metadata</code> .
...	Additional options, only used when <code>legacy = TRUE</code> . User may pass an argument called <code>questions</code> , a vector containing the names of questions for which you want to return metadata.

Value

A list containing survey description metadata. The contents of the returned list depend on `elements`.

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()

# Get description for a survey
descrip <- fetch_description(surveyID = surveys$id[6])

# Get metadata with specific elements
descrip_specific <- fetch_description(
  surveyID = id,
  elements = c("questions", "flow")
)

## End(Not run)
```

fetch_distributions *Download distribution data for a survey from Qualtrics*

Description

Download distribution data for a survey from Qualtrics

Usage

```
fetch_distributions(surveyID)
```

Arguments

surveyID String. Unique survey ID for the distribution data you want to download.

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>"
)

surveys <- all_surveys()
distributions <- fetch_distributions(surveys$id[1])

## End(Not run)
```

fetch_id	<i>Fetch a unique Qualtrics survey ID based on survey name in the Qualtrics UI</i>
----------	--

Description

Fetch a unique Qualtrics survey ID based on survey name in the Qualtrics UI

Usage

```
fetch_id(.data, survey_name)
```

Arguments

.data Data frame of active surveys created by the function [all_surveys\(\)](#).

survey_name Name of the survey as it appears in the Qualtrics UI. Must be unique to be passed to `fetch_id()`.

Details

Survey names in the Qualtrics platform are not required to be unique, but the `survey_name` argument for this function *must* be unique.

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 surveyID for a unique survey
my_id <- fetch_id(surveys, "Unique Survey Name")

## End(Not run)
```

fetch_mailinglist *Download a mailing list from Qualtrics*

Description

Download a mailing list from Qualtrics

Usage

```
fetch_mailinglist(mailinglistID)
```

Arguments

mailinglistID String. Unique ID for the mailing list you want to download. Returned as id by the [all_mailinglists](#) function.

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 mailing lists
mailinglists <- all_mailinglists()

# Retrieve a single mailing list
mailinglist <- fetch_mailinglist(mailinglists$id[1])

## End(Not run)
```

 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 = deprecated(),
  start_date = NULL,
  end_date = NULL,
  unanswer_recode = NULL,
  unanswer_recode_multi = unanswer_recode,
  include_display_order = TRUE,
  limit = NULL,
  include_questions = NULL,
  save_dir = NULL,
  force_request = FALSE,
  verbose = TRUE,
  label = TRUE,
  convert = TRUE,
  import_id = FALSE,
  time_zone = NULL,
  breakout_sets = TRUE,
  add_column_map = TRUE,
  add_var_labels = TRUE,
  col_types = NULL,
  ...
)
```

Arguments

surveyID	String. Unique ID for the survey you want to download. Returned as id by the all_surveys function.
last_response	Deprecated.
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	Integer. Recode seen but unanswered questions with an integer-like value, such as 999. Defaults to NULL.

unanswer_recode_multi	Integer. Recode seen but unanswered multi-select questions with an integer-like value, such as 999. Defaults to value for unanswer_recode.
include_display_order	Display order information (such as for surveys with randomization).
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.
import_id	Logical. If TRUE, use Qualtrics import IDs instead of question IDs as column names. Will also alter names in the column map, if used. Defaults to FALSE.
time_zone	String. A local timezone to determine response date values. Defaults to NULL which corresponds to UTC time. See " Dates and Times " from Qualtrics for more information on format.
breakout_sets	Logical. If TRUE, then the <code>fetch_survey</code> function will split multiple choice question answers into columns. If FALSE, each multiple choice question is one column. Defaults to TRUE.
add_column_map	Logical. If TRUE, then a column map data frame will be added as an attribute to the main response data frame. This column map captures Qualtrics-provided metadata associated with the response download, such as an item description and internal ID's. Defaults to TRUE.
add_var_labels	Logical. If TRUE, then the item description from each variable (equivalent to the one in the column map) will be added as a "label" attribute using <code>set_label</code> . Useful for reference as well as cross-compatibility with other stats packages (e.g., Stata, see documentation in <code>sjlabelled</code>). Defaults to TRUE.
col_types	Optional. This argument provides a way to manually overwrite column types that may be incorrectly guessed. Takes a <code>cols</code> specification. See example below and <code>cols</code> for formatting details. Defaults to NULL. Overwritten by <code>convert = TRUE</code> .
...	Optional arguments, such as a <code>fileEncoding</code> (see <code>fileEncoding</code> argument in <code>read_survey</code>) to import your survey using a specific encoding.

See Also

See <https://api.qualtrics.com/> 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 = 999,
  verbose = TRUE,
  # Manually override EndDate to be a character vector
  col_types = readr::cols(EndDate = readr::col_character())
)

## End(Not run)
```

getSurvey

Download a survey and import it into R

Description

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

Usage

```
getSurveyQuestions(...)
```

Arguments

... All arguments for `survey_questions`

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

Description

This function is 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 = NULL, ...)
```

Arguments

surveyID	A string. Unique ID for the survey you want to download. Returned as "id" by the all_surveys function.
get	A character vector containing any of the following: "metadata", "questions", "responsecounts", "blocks", "flow", "embedded_data", or "comments". Will return included elements. By default, the function returns the "metadata", "questions", and "responsecounts" elements. See examples below for more information.
...	Additional options. User may pass an argument called questions, 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>"
)

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

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

# Get metadata with specific elements
md_specific <- metadata(
  surveyID = id,
  get = c("flow")
)

# Get specific question metadata
question_specific <- metadata(
  surveyID = id,
```



```

    get = c("questions"),
    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 deprecated; use [qualtrics_api_credentials](#) instead.

Usage

```
qualtRicsConfigFile(...)
```

Arguments

... All arguments for qualtRicsConfigFile

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

api_key	The API key provided to you from Qualtrics formatted in quotes. Learn more about Qualtrics API keys at https://api.qualtrics.com/
base_url	The institution-specific base URL for your Qualtrics account, formatted in quotes. Find your base URL at https://api.qualtrics.com/
overwrite	If TRUE, will overwrite existing Qualtrics credentials that you already have in your .Renvirom file.
install	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  
readRenvirom("~/Renvirom")  
# 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_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

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

`readSurvey`

Read a CSV file exported from Qualtrics

Description

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

*Read a CSV file exported from Qualtrics***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,
  import_id = FALSE,
  time_zone = NULL,
  legacy = FALSE,
  add_column_map = TRUE,
  add_var_labels = TRUE,
  col_types = NULL
)
```

Arguments

file_name	String. A CSV data file.
strip_html	Logical. If TRUE, then remove HTML tags. Defaults to TRUE.
import_id	Logical. If TRUE, use Qualtrics import IDs instead of question IDs as column names. Defaults to FALSE.
time_zone	String. A local timezone to determine response date values. Defaults to NULL which corresponds to UTC time. See " Dates and Times " from Qualtrics for more information on format.
legacy	Logical. If TRUE, then import "legacy" format CSV files (as of 2017). Defaults to FALSE.
add_column_map	Logical. If TRUE, then a column map data frame will be added as an attribute to the main response data frame. This column map captures Qualtrics-provided metadata associated with the response download, such as an item description and internal ID's. Defaults to TRUE.
add_var_labels	Logical. If TRUE, then the item description from each variable (equivalent to the one in the column map) will be added as a "label" attribute using set_label . Useful for reference as well as cross-compatibility with other stats packages (e.g., Stata, see documentation in sjlabelled). Defaults to TRUE.
col_types	Optional. This argument provides a way to manually overwrite column types that may be incorrectly guessed. Takes a cols specification. See example below and cols for formatting details. Defaults to NULL.

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)

# Example changing column type
file <- system.file("extdata", "sample.csv", package = "qualtrics")
# Force EndDate to be a string
df <- read_survey(file, col_types = readr::cols(EndDate = readr::col_character()))
```

survey_questions

Retrieve a data frame containing question IDs and labels

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 [all_surveys](#) function.

See Also

See <https://api.qualtrics.com/> 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, question_meta)
```

Arguments

data	Imported Qualtrics survey
question_meta	Question metadata

Index

[all_mailinglists](#), [2](#), [11](#)
[all_surveys](#), [3](#), [5](#), [7](#), [9](#), [12](#), [15](#), [16](#), [21](#)
[all_surveys\(\)](#), [10](#)

[check_for_warnings](#), [4](#)
[check_params](#), [4](#)
[cols](#), [13](#), [20](#)
[column_map](#), [5](#)
[construct_header](#), [6](#)
[create_raw_payload](#), [6](#)

[download_qualtrics_export](#), [7](#)

[extract_colmap](#), [8](#)

[fetch_description](#), [8](#)
[fetch_distributions](#), [9](#)
[fetch_id](#), [10](#)
[fetch_mailinglist](#), [11](#)
[fetch_survey](#), [7](#), [8](#), [12](#), [13](#), [14](#)

[getSurvey](#), [14](#)
[getSurveyQuestions](#), [15](#)
[getSurveys](#), [15](#)

[infer_data_types](#), [15](#)

[metadata](#), [9](#), [16](#)

[qualtrics_api_credentials](#), [17](#), [17](#)
[qualtrics_response_codes](#), [4](#), [19](#)
[qualtRicsConfigFile](#), [17](#)

[read_survey](#), [13](#), [19](#), [20](#)
[readRDS](#), [13](#)
[readSurvey](#), [19](#)

[set_label](#), [13](#), [20](#)
[survey_questions](#), [15](#), [21](#)

[wrapper_mc](#), [22](#)