

Package ‘rsurveycto’

February 12, 2026

Type Package

Title Interact with Data on 'SurveyCTO'

Version 0.2.5

Description 'SurveyCTO' is a platform for mobile data collection in offline settings.

The 'rsurveycto' R package uses the 'SurveyCTO' REST API

<<https://docs.surveycto.com/05-exporting-and-publishing-data/05-api-access/01.api-access.html>>

to read datasets and forms from a 'SurveyCTO' server into R as 'data.table's
and to download file attachments. The package also has limited support to
write datasets to a server.

URL <https://agency-fund.github.io/rsurveycto/>,

<https://github.com/agency-fund/rsurveycto>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports checkmate (>= 2.3.2), cli (>= 3.6.3), curl (>= 5.2.1),
data.table (>= 1.15.4), glue (>= 1.7.0), httr (>= 1.4.7),
jsonlite (>= 1.8.8), lifecycle (>= 1.0.4), readxl (>= 1.4.3),
rlang (>= 1.1.4), vctrs (>= 0.6.5), withr (>= 3.0.1)

Suggests knitr, rmarkdown, testthat (>= 3.2.1.1)

Config/testthat/edition 3

NeedsCompilation no

Author Jake Hughey [aut, cre],
Robert On [aut]

Maintainer Jake Hughey <jake@agency.fund>

Repository CRAN

Date/Publication 2026-02-12 08:20:12 UTC

Contents

drop_empties	2
scto_auth	3
scto_get_attachments	4
scto_get_form_definitions	5
scto_get_form_metadata	6
scto_get_form_responses	7
scto_meta	8
scto_quiet	8
scto_read	9
scto_unnest_form_definitions	10
scto_write	11
Index	13

drop_empties	<i>Drop empty columns from a data.table</i>
--------------	---

Description

An empty column is one whose only values are NA or "".

Usage

```
drop_empties(d)
```

Arguments

d data.table.

Value

d modified by reference, invisibly.

Examples

```
library('data.table')
d = data.table(w = 3:4, x = c('', 'foo'), y = c(NA, NA), z = c(NA, ''))
drop_empties(d)
```

scto_auth

Authenticate with a SurveyCTO server

Description

SurveyCTO's API supports basic authentication using a username and password. Make sure the user is assigned a role with permission to download data ("data manager" or greater) and "Allow server API access" is enabled.

Usage

```
scto_auth(  
  auth_file = NULL,  
  servername = NULL,  
  username = NULL,  
  password = NULL,  
  validate = TRUE  
)
```

Arguments

auth_file	String indicating path to a text file containing the server name on the first line, username on the second, and password on the third. Other arguments are only used if auth_file is NULL.
servername	String indicating name of the SurveyCTO server.
username	String indicating username for the SurveyCTO account.
password	String indicating password for the SurveyCTO account.
validate	Logical indicating whether to validate credentials by calling scto_meta() . Should only be set to FALSE for debugging.

Value

scto_auth object for an authenticated SurveyCTO session.

Examples

```
## Not run:  
# preferred approach  
auth = scto_auth('scto_auth.txt')  
  
# alternate approach  
auth = scto_auth('my_server', 'my_user', 'my_pw', auth_file = NULL)  
  
## End(Not run)
```

scto_get_attachments *Fetch file attachments from a SurveyCTO server*

Description

This function can download encrypted and unencrypted files attached to forms.

Usage

```
scto_get_attachments(  
  auth,  
  urls,  
  output_dir,  
  private_key = NULL,  
  overwrite = TRUE  
)
```

Arguments

auth	scto_auth() object.
urls	Character vector of API URLs for file attachments. Will typically be derived from a column of a data table returned by scto_read() . Can contain missing values.
output_dir	String indicating path to directory in which to save files.
private_key	String indicating path to private key file. Only needs to be non-NULL to decrypt encrypted file attachments.
overwrite	Logical indicating whether to overwrite existing files.

Value

A character vector of file names of the same length as `urls`, with NA for missing or invalid URLs.

Examples

```
## Not run:  
auth = scto_auth('scto_auth.txt')  
form_data = scto_read(auth, 'my_form', 'form')  
filenames = scto_get_attachments(auth, form_data[['my_attachment']], '.')  
  
## End(Not run)
```

`scto_get_form_definitions`*Fetch deployed form definitions from a SurveyCTO server*

Description

[Superseded]

This function fetches definitions for currently deployed forms. It has been superseded in favor of [scto_get_form_metadata\(\)](#), which fetches metadata, including definitions, for deployed and previous versions of forms.

Usage

```
scto_get_form_definitions(auth, form_ids = NULL, simplify = TRUE)
```

Arguments

<code>auth</code>	scto_auth() object.
<code>form_ids</code>	Character vector indicating the form ids. NULL indicates all forms.
<code>simplify</code>	Logical indicating whether to return the definition for one form as a simple list instead of a named, nested list.

Value

If `simplify` is TRUE and getting one form definition, a list. Otherwise a named list of lists containing the definition for each form.

Examples

```
## Not run:
auth = scto_auth('scto_auth.txt')
form_def = scto_get_form_definitions(auth, 'my_form')
form_defs = scto_get_form_definitions(auth)

## End(Not run)
```

`scto_get_form_metadata`*Fetch form metadata from a SurveyCTO server*

Description

This function fetches metadata, including form definitions, for deployed and previous versions of one or more forms.

Usage

```
scto_get_form_metadata(  
  auth,  
  form_ids = NULL,  
  deployed_only = FALSE,  
  get_defs = TRUE,  
  def_dir = NULL  
)
```

Arguments

<code>auth</code>	scto_auth() object.
<code>form_ids</code>	Character vector indicating the form ids. NULL indicates all forms.
<code>deployed_only</code>	Logical indicating whether to fetch metadata for all versions of each form, or only for the deployed version.
<code>get_defs</code>	Logical indicating whether to fetch form definitions.
<code>def_dir</code>	String indicating directory in which to save the form definitions as Excel files.

Value

A `data.table` with one row per form (and per version, if `deployed_only` is FALSE). Definitions are returned as nested `data.tables`, which can be unnested using [scto_unnest_form_definitions\(\)](#).

Examples

```
## Not run:  
auth = scto_auth('scto_auth.txt')  
form_metadata = scto_get_form_metadata(auth, 'my_form')  
form_defs = scto_unnest_form_definitions(form_metadata)  
  
## End(Not run)
```

scto_get_form_responses

Read form data from a SurveyCTO server and convert to long format

Description

This function converts data from `scto_read()` to long format, then joins the data with the "survey" and "choices" components of the form metadata.

Usage

```
scto_get_form_responses(
  auth,
  form_id,
  id_cols = c("KEY", "formdef_version", "CompletionDate", "SubmissionDate", "starttime",
    "endtime", "review_status", "review_quality"),
  exclude_cols = "instanceID",
  ...
)
```

Arguments

<code>auth</code>	<code>scto_auth()</code> object.
<code>form_id</code>	String indicating id of the form.
<code>id_cols</code>	Character vector of data columns to keep in the result, will be passed as <code>id.vars</code> to <code>data.table::melt()</code> .
<code>exclude_cols</code>	Character vector of data columns to exclude from the result.
<code>...</code>	Additional arguments passed to <code>scto_read()</code> .

Value

A `data.table` with one row per response, i.e., one row per form submission per field. If necessary to avoid ambiguity, columns from the "survey" part of the form definition are prefixed with `field_`, while columns from the "choices" part of the form definition are prefixed with `choice_`. Original column names from data returned by `scto_read()` are in the `submission_field_name` column, while corresponding values are in the `field_value` column. Data can be converted back to wide format as desired using `data.table::dcast()` or `tidyr::pivot_wider()`.

Examples

```
## Not run:
auth = scto_auth('scto_auth.txt')
responses = scto_get_form_responses(auth, 'my_form')

## End(Not run)
```

scto_meta	<i>Read metadata from a SurveyCTO server</i>
-----------	--

Description

These functions read metadata from a SurveyCTO server.

Usage

```
scto_meta(auth)

scto_catalog(auth)
```

Arguments

auth [scto_auth\(\)](#) object.

Value

scto_meta() returns a nested list of metadata related to forms, datasets, groups, and publishing information. scto_catalog() returns a `data.table` with one row per form or dataset.

Examples

```
## Not run:
auth = scto_auth('scto_auth.txt')
metadata = scto_meta(auth)
catalog = scto_catalog(auth)

## End(Not run)
```

scto_quiet	<i>Suppress or permit messages from rsurveycto</i>
------------	--

Description

By default, rsurveycto prints messages to the console. To suppress them, set the `rsurveycto_quiet` option to TRUE or use this function.

Usage

```
scto_quiet(quiet = NULL)
```

Arguments

quiet A logical indicating whether to suppress messages, or NULL.

Value

If quiet is NULL, the current value of the rsurveycto_quiet option. Otherwise, the previous value of the rsurveycto_quiet option invisibly.

Examples

```
options(rsurveycto_quiet = TRUE)
scto_quiet()
scto_quiet(FALSE)
```

scto_read	<i>Read data from a SurveyCTO server</i>
-----------	--

Description

This function can read datasets and forms.

Usage

```
scto_read(
  auth,
  ids = NULL,
  start_date = as.POSIXct("1900-01-01", tz = "UTC"),
  review_status = "approved",
  private_key = NULL,
  drop_empty_cols = FALSE,
  convert_datetime = c("CompletionDate", "SubmissionDate", "starttime", "endtime"),
  datetime_format = "%b %e, %Y %I:%M:%S %p",
  simplify = TRUE
)
```

Arguments

auth	scto_auth() object.
ids	Character vector indicating ids of the datasets and/or forms. NULL indicates all datasets and forms.
start_date	Date-time or something coercible to a date-time indicating the earliest date-time (UTC timezone) for which to fetch data. Only used for forms. Use with caution, because fields that are deleted prior to start_date will not show up, even if submissions prior to start_date have data for those fields.
review_status	String or character vector indicating which submissions to fetch. Possible values are "approved", "pending", "rejected", or any combination of the three. Only used for forms.
private_key	String indicating path to private key file. Only needs to be non-NULL to read encrypted form data.

drop_empty_cols	Logical indicating whether to drop columns that contain only NA or only an empty string.
convert_datetime	Character vector of column names in the data for which to convert strings to datetimes (POSIXct). Use NULL to not convert any columns to datetimes.
datetime_format	String indicating format of datetimes from SurveyCTO. See strptime() .
simplify	Logical indicating whether to return only a <code>data.table</code> instead of a list of <code>data.tables</code> if reading one form or dataset.

Value

If `simplify` is `TRUE` and reading one form or dataset, a `data.table`. Otherwise a named list of `data.tables`, one for each form and dataset, along with a `data.table` named `".catalog"` from `scto_catalog()`.

See Also

[scto_get_form_responses\(\)](#)

Examples

```
## Not run:
auth = scto_auth('scto_auth.txt')
form_data = scto_read(auth, 'my_form')
all_data = scto_read(auth)

## End(Not run)
```

```
scto_unnest_form_definitions
```

Unnest previously fetched form definitions

Description

This function unnests form definitions, e.g., from multiple versions of a form, which can make it easier to map values to labels in a later step.

Usage

```
scto_unnest_form_definitions(form_metadata, by_form_id = TRUE)
```

Arguments

form_metadata	<code>data.table</code> returned by scto_get_form_metadata() .
by_form_id	Logical indicating whether to unnest definitions of multiple versions of a given form (default), or to unnest definitions of all forms together.

Value

If `by_form_id` is `TRUE`, a `data.table` of `data.tables` for the survey, choices, and settings components of the form definitions. Otherwise a list of `data.tables`.

Examples

```
## Not run:
auth = scto_auth('scto_auth.txt')
form_metadata = scto_get_form_metadata(auth, 'my_form')
form_defs = scto_unnest_form_definitions(form_metadata)

## End(Not run)
```

scto_write

*Write data to a SurveyCTO server***Description****[Experimental]**

This function updates an existing dataset using an unofficial API endpoint. For this to work, the SurveyCTO user must have permissions "Can edit server datasets" and "Can modify or delete server dataset data".

Usage

```
scto_write(
  auth,
  data,
  dataset_id,
  dataset_title = dataset_id,
  append = FALSE,
  fill = FALSE
)
```

Arguments

<code>auth</code>	<code>scto_auth()</code> object.
<code>data</code>	<code>data.frame</code> to upload.
<code>dataset_id</code>	String indicating id of existing dataset.
<code>dataset_title</code>	String indicating title of dataset. Will replace the existing title, regardless of <code>append</code> .
<code>append</code>	Logical indicating whether to append or replace the dataset.
<code>fill</code>	Logical indicating whether to implicitly fill missing columns with NA, i.e., whether to allow a mismatch between columns of the existing dataset and columns of data. Only used if <code>append</code> is <code>TRUE</code> .

Value

A list with elements:

- data_old: A `data.table` of the previous version of the dataset.
- response: An object of class `httr::response()` from the API call.

Examples

```
## Not run:  
auth = scto_auth('scto_auth.txt')  
r = scto_write(auth, data, 'my_dataset', 'My Dataset')  
  
## End(Not run)
```

Index

`data.table::dcast()`, [7](#)
`data.table::melt()`, [7](#)
`drop_emptyies`, [2](#)

`httr::response()`, [12](#)

`scto_auth`, [3](#)
`scto_auth()`, [4–9](#), [11](#)
`scto_catalog(scto_meta)`, [8](#)
`scto_get_attachments`, [4](#)
`scto_get_form_definitions`, [5](#)
`scto_get_form_metadata`, [6](#)
`scto_get_form_metadata()`, [5](#), [10](#)
`scto_get_form_responses`, [7](#)
`scto_get_form_responses()`, [10](#)
`scto_meta`, [8](#)
`scto_meta()`, [3](#)
`scto_quiet`, [8](#)
`scto_read`, [9](#)
`scto_read()`, [4](#), [7](#)
`scto_unnest_form_definitions`, [10](#)
`scto_unnest_form_definitions()`, [6](#)
`scto_write`, [11](#)
`strptime()`, [10](#)

`tidyr::pivot_wider()`, [7](#)