

Package ‘REDCapExporter’

May 15, 2020

Title Automated Construction of R Data Packages from REDCap Projects

Version 0.2.1

Description Export all data, including metadata, from a REDCap (Research Electronic Data Capture) Project via the REDCap API
<<https://projectredcap.org/wp-content/resources/REDCapTechnicalOverview.pdf>>.
The exported (meta)data will be processed and formatted into a stand alone R data package which can be installed and shared between researchers. Several default reports are generated as vignettes in the resulting package.

License GPL-2

Encoding UTF-8

URL <https://github.com/dewittpe/REDCapExporter>

Language en-us

LazyData true

Depends R (>= 3.5.0)

Imports curl, data.table, keyring, lubridate

Suggests devtools, fs, knitr, magrittr, qwraps2 (> 0.4.1), rmarkdown, rjson, secret, testthat

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no

Author Peter DeWitt [aut, cre]

Maintainer Peter DeWitt <peter.dewitt@cuanschutz.edu>

Repository CRAN

Date/Publication 2020-05-15 17:10:13 UTC

R topics documented:

as	2
build_r_data_package	3
col_type	4

example_data	5
export_content	6
export_core	7
format_record	8
read_text	9
REDCapExporter	9
REDCapExporter_keyring	10
write_description_file	11

Index	12
--------------	-----------

as	<i>as.*</i>
----	-------------

Description

Coerce REDCapExporter objects to data.frame.

Usage

```
## S3 method for class 'rcer_raw_metadata'
as.data.frame(x, ...)
```

```
## S3 method for class 'rcer_raw_record'
as.data.frame(x, ...)
```

```
## S3 method for class 'rcer_raw_project'
as.data.frame(x, ...)
```

```
## S3 method for class 'rcer_raw_user'
as.data.frame(x, ...)
```

```
## S3 method for class 'rcer_raw_metadata'
as.data.table(x, ...)
```

```
## S3 method for class 'rcer_raw_record'
as.data.table(x, ...)
```

```
## S3 method for class 'rcer_raw_project'
as.data.table(x, ...)
```

```
## S3 method for class 'rcer_raw_user'
as.data.table(x, ...)
```

Arguments

x any R object.
 ... additional arguments to be passed to or from methods.

Details

These functions are S3 methods for `as.data.frame` for the raw exports from the REDCap API.

Examples

```
data("avs_raw_record")

avs_record <- as.data.frame(avs_raw_record)

str(avs_record)
```

build_r_data_package *Build R Data Package*

Description

Build a R Data Package from the core contents of a REDCap Project.

Usage

```
build_r_data_package(x, ...)

## S3 method for class 'rcer_rccore'
build_r_data_package(x, path = NULL, author_roles = NULL, verbose = TRUE, ...)

## Default S3 method:
build_r_data_package(
  x,
  uri = NULL,
  token = NULL,
  format = NULL,
  path = NULL,
  author_roles = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

<code>x</code>	a <code>rcer_rccore</code> object
<code>...</code>	arguments passed to other methods
<code>path</code>	Path where the exported project source will be created/overwritten.
<code>author_roles</code>	a list naming specific roles for each user id found in the user table from an exported project. By default all users will be contributors ('ctb'). You will need to define a author/creator.

verbose	provide messages to tell the user what is happening
uri	The URI for the REDCap API. If NULL (default) the value <code>Sys.getenv("REDCap_API_URI")</code> is used.
token	The API token for the project you want to export from. If NULL (default) the value <code>Sys.getenv("REDCap_API_TOKEN")</code> is used.
format	The format to return. If NULL (default) the value <code>Sys.getenv("REDCap_API_format")</code> is used.

Details

To export the data from a REDCap project you will need to have an API Token. Remember, the token is the equivalent of a username and password. As such you should not list the token in plain text. Several alternative methods for passing the token to this method will be provided in examples and vignettes. We strongly encourage the use of the package secret <https://cran.r-project.org/package=secret> to build vaults to store tokens locally.

The initial export will consist of four pieces of data, the user data, metadata, project info, and records.

Examples

```
## Please read the vignette for examples:
## vignette(topic = "export", package = "REDCapExporter")

library(REDCapExporter)
# avs_raw_core <- export_core()
data(avs_raw_core)
tmppth <- tempdir()
build_r_data_package(avs_raw_core, tmppth, author_roles = list(dewittp = c("cre", "aut")))
fs::dir_tree(tmppth)
```

col_type

Column Types

Description

Define a type for each column of the records for a REDCap project based on the metadata for the project.

Usage

```
col_type(x, factors = TRUE, ...)
```

Arguments

x	a <code>rcer_metadata</code> or <code>rcer_raw_metadata</code> object
factors	If TRUE (default) then variables reported via drop-down lists and radio buttons are set up to be factors. If FALSE, then the column type will be character.
...	not currently used

Value

a rcer_col_type object

Examples

```
data("avs_raw_metadata")
col_type(avs_raw_metadata)
```

example_data

Raw Exports From an Example REDCap Project

Description

These data sets are the results of calling [export_content](#). An API token is required to reproduce these calls. No such token will be provided publicly, so these data sets are provided so end users can run examples for other tools provided in the REDCapExporter package.

Usage

avs_raw_project_info

avs_raw_metadata

avs_raw_user

avs_raw_record

avs_raw_core

Format

An object of class rcer_raw_project (inherits from character) of length 1.

An object of class rcer_raw_metadata (inherits from character) of length 1.

An object of class rcer_raw_user (inherits from character) of length 1.

An object of class rcer_raw_record (inherits from character) of length 1.

An object of class rcer_rccore of length 4.

Details

avs_raw_project_info provides meta data about the project itself.

avs_raw_metadata is the data dictionary for the REDCap Project. This information can be used with [format_record](#) to build a data.frame that is ready for analysis.

avs_raw_user REDCap Project user table.

avs_raw_record REDCap Project records, i.e., 'the data.'

Examples

```
## Not run:
avs_raw_project_info <- export_content(content = "project", format = "csv")
avs_raw_metadata <- export_content(content = "metadata", format = "csv")
avs_raw_user <- export_content(content = "user", format = "csv")
avs_raw_record <- export_content(content = "record", format = "csv")
avs_raw_core <- export_core(format = "csv")

## End(Not run)

data(avs_raw_project_info)
data(avs_raw_user)
data(avs_raw_metadata)
data(avs_raw_record)
data(avs_raw_core)

str(avs_raw_project_info)
str(avs_raw_user)
str(avs_raw_metadata)
str(avs_raw_record)
str(avs_raw_core)

avs <- format_record(avs_raw_record, avs_raw_metadata)
str(avs)
```

export_content

Export Content

Description

Export specific data elements from REDCap

Usage

```
export_content(content, uri = NULL, token = NULL, format = NULL, ...)
```

Arguments

content	The element to export, see Details.
uri	The URI for the REDCap API. If NULL (default) the value <code>Sys.getenv("REDCap_API_URI")</code> is used.
token	The API token for the project you want to export from. If NULL (default) the value <code>Sys.getenv("REDCap_API_TOKEN")</code> is used.
format	The format to return. If NULL (default) the value <code>Sys.getenv("REDCap_API_format")</code> is used.
...	additional arguments passed to handle_setform .

Details

The content and format arguments are used to control the specific items to be exported, and in what format. ****Review the API documentation****

The uri, token, and format arguments are set to NULL by default and will look to the `Sys.getenv("REDCap_API_URI")`, `Sys.getenv("REDCap_API_TOKEN")`, and `Sys.getenv("REDCap_API_format")`, respectively, to define the values if not explicitly done so by the end user.

Value

The raw return from the REDCap API with the class `rceer_raw_<content>`.

Examples

```
# A reproducible example would require a REDCap project, accessible via an
# API token. An example of the return from these calls are provided as data
# with this package.

# avs_raw_metadata <- export_content(content = "metadata")
data(avs_raw_metadata)
str(avs_raw_metadata)
```

 export_core

Export Core

Description

Export Core Contents of a REDCap Project.

Usage

```
export_core(uri = NULL, token = NULL, format = NULL, verbose = TRUE, ...)
```

Arguments

uri	The URI for the REDCap API. If NULL (default) the value <code>Sys.getenv("REDCap_API_URI")</code> is used.
token	The API token for the project you want to export from. If NULL (default) the value <code>Sys.getenv("REDCap_API_TOKEN")</code> is used.
format	The format to return. If NULL (default) the value <code>Sys.getenv("REDCap_API_format")</code> is used.
verbose	provide messages to tell the user what is happening
...	not currently used

Value

A rcer_rccore object: a list with the project info, metadata, user table, and records, all in a "raw" format direct from the API.

Examples

```
# A reproducible example would require a REDCap project, accessible via an
# API token. An example of the return from these calls are provided as data
# with this package.
```

```
# avs_raw_core <- export_core()
```

```
data(avs_raw_core)
str(avs_raw_core)
```

format_record

Format Record

Description

Use REDCap project metadata to build a well formatted data.frame for the record.

Usage

```
format_record(
  record,
  metadata = NULL,
  col_type = NULL,
  class = "data.table",
  ...
)
```

Arguments

record	a rcer_raw_record object.
metadata	a rcer_metadata or rcer_raw_metadata object. Will be ignored if col_type is defined.
col_type	a rcer_col_type object.
class	return either a data.frame or data.table
...	other arguments passed to col_type

Value

A data.frame or data.table

Examples

```
data("avs_raw_metadata")
data("avs_raw_record")

avs <- format_record(avs_raw_record, avs_raw_metadata)
```

read_text	<i>Read Text</i>
-----------	------------------

Description

Read raw REDCap API return. Built to parse csv or json.

Usage

```
read_text(x, class = "data.frame")
```

Arguments

x	the raw return from the API call to REDCap
class	"data.frame" (default) or "data.table"

Details

This is a non-exported function and not expected to be called by the end user. Used by the `as.data.frame` methods.

Value

a `data.frame`

REDCapExporter	<i>REDCapExporter</i>
----------------	-----------------------

Description

Automated Construction of R Data Packages from REDCap Projects

Details

Use the REDCap (Research Electronic Data Capture) API to access (meta)data from a project and format for immediate use or as part of the source code for a R Data package.

Please read the vignette for examples and use cases.

```
vignette(topic = "export", package = "REDCapExporter")
```

`REDCapExporter_keyring`*Set up and use of a Keyring for REDCap API Tokens*

Description

Tools for checking for, and setting up, a file based keyring for storing REDCap API tokens.

Usage

```
REDCapExporter_keyring_check(keyring = "REDCapExporter", password = NULL)
```

```
REDCapExporter_add_api_token(  
  project,  
  keyring = "REDCapExporter",  
  user = NULL,  
  password = NULL,  
  overwrite = FALSE  
)
```

```
REDCapExporter_get_api_token(  
  project,  
  keyring = "REDCapExporter",  
  user = NULL,  
  password = NULL  
)
```

Arguments

<code>keyring</code>	a character vector identifying the name of the keyring, defaults to "REDCapExporter"
<code>password</code>	This is the password for the keyring. The default is an empty password.
<code>project</code>	the name of the REDCap project the API token is identified by.
<code>user</code>	user name to associate the token with. Defaults to <code>Sys.info()[["user"]]</code> .
<code>overwrite</code>	logical, if TRUE overwrite the existing token.

Value

`REDCapExporter_keyring_check` returns TRUE, invisibly, as does `REDCapExporter_add_api_token`. `REDCapExporter_get_api_token` returns the token invisibly as not to print the value to the console by default. Still, be careful with your token.

See Also

```
vignette(topic = "api", package = "REDCapExporter")
```

Examples

```
## Not run:
REDCapExporter_keyring_check()
REDCapExporter_add_api_token("Project1")
Sys.setenv(REDCap_API_TOKEN = REDCapExporter_get_api_token("Project1"))

## End(Not run)
```

write_description_file

Write Description File

Description

Create the DESCRIPTION file for the R Data package based on an Exported REDCap Project

Usage

```
write_description_file(access_time, user, roles, project_info, path)

write_authors(user, roles = NULL)
```

Arguments

access_time	The Sys.time() when the API calls were made
user	User(s), as noted in the REDCap project meta data. This parameter is singular as it refers to the "user" content one can access from the REDCap API.
roles	roles the user hold with respect to the R data package. These roles have no relationship to REDCap roles.
project_info	project metadata
path	path to the root for the generated R data package.

Details

These is a non-exported function and is not expected to be called by the end user.

write_description_file creates the DESCRIPTION file for the exported R data package and write_authors creates the "Authors@R" field of the DESCRIPTION based on the "user" data extracted from the REDCap project.

Index

*Topic **datasets**

- example_data, 5

- as, 2
- as.data.frame, 3
- avs_raw_core (example_data), 5
- avs_raw_metadata (example_data), 5
- avs_raw_project_info (example_data), 5
- avs_raw_record (example_data), 5
- avs_raw_user (example_data), 5

- build_r_data_package, 3

- col_type, 4, 8

- example_data, 5
- export_content, 5, 6
- export_core, 7

- format_record, 5, 8

- handle_setform, 6

- read_text, 9
- REDCapExporter, 9
- REDCapExporter_add_api_token
 (REDCapExporter_keyring), 10
- REDCapExporter_get_api_token
 (REDCapExporter_keyring), 10
- REDCapExporter_keyring, 10
- REDCapExporter_keyring_check
 (REDCapExporter_keyring), 10

- write_authors (write_description_file),
 11
- write_description_file, 11