# Package 'rccola'

January 20, 2022

**Type** Package

**Title** Safely Manage API Keys and Load Data from a REDCap or Other Source

**Version** 1.0.2

**Author** Shawn Garbett [aut, cre],
Hui Wu [aut],
Cole Beck [aut]

**Maintainer** Shawn Garbett <Shawn.Garbett@vumc.org>

**Description** The handling of an API key (misnomer for password) for protected data can be difficult. This package provides secure convenience functions for entering / handling API keys and pulling data directly into memory. By default it will load from REDCap instances, but other sources are injectable via inversion of control.

**License** GPL-3

**Encoding** UTF-8

**Imports** redcapAPI, getPass, yaml, keyring (>= 1.3.0)

**URL** https://github.com/spgarbet/rccola

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-01-20 16:12:42 UTC

## R topics documented:

drinkREDCap *Provide API_KEYs to function (defaults to load from REDCap) and load data into memory.*

## Description

The first thing it does is check for a yaml config file of the same name as the current directory with a .yml extension one level above. This is intended for production environments where the API_KEY must be stored in a file. If this yaml exists, then it expects this file to contain ‘apiUrl‘ and ‘apiKeys‘. ‘apiUrl‘ should be a string with the URL of the REDCap instance. ‘apiKeys‘ should be a list of variable name keys with values that are their actual REDCap API_KEY.

Next it will use an api environment in memory to keep api_keys. If one is knitting with parameters, it will request and store these keys in memory. Otherwise it will request the user enter each key using getPass and store it in memory.

IMPORTANT: Make sure that R is set to NEVER save workspace to .RData as this is the equivalent of writing the API_KEY to a local file in clear text.

## Usage

```
drinkREDCap(
  variables,
  keyring = NULL,
  envir = NULL,
  forms = NULL,
  FUN = sipREDCap,
  config = "auto",
  assign = TRUE,
  passwordFUN = getPass::getPass,
  ...
)

loadFromRedcap(
  variables,
  keyring = NULL,
  envir = NULL,
  forms = NULL,
  FUN = sipREDCap,
  config = "auto",
  assign = TRUE,
  passwordFUN = getPass::getPass,
  ...
)
```

### Arguments

| | |
|---|---|
| variables | character vector. A list of strings that define the variables with associated API_KEYs to load into memory. |
| keyring | character. Potential keyring, not used by default. |
| envir | environment. The target environment for the data. Defaults to .Global |
| forms | list. A list of forms. Keys are the variable(api_key), each key can contain a vector of forms. The output variable is now the <variable>.<form> |
| FUN | function. the function to call. It must have a key argument. If forms are used it should have a forms argument as well. The default is to call sipREDCap which is a proxy for exportRecords. |
| config | string. Defaults to 'auto'. If set to NULL no configuration file is searched for. If set to anything but 'auto', that will be the config file override that is used if it exists instead of searching for the ../<basename>.yml. |
| assign | logical. Does the function write back the variable to envir or not. Defaults to TRUE. |
| passwordFUN | function. Function to get the password for the keyring. Defaults to getPass::getPass(). |
| ... | Additional arguments passed to FUN. |

### Details

An older loadFromRedcap function maps to this for backward compatibility.

### Value

Nothing

### Examples

```
## Not run:
  drinkREDCap("database", "myproject")

## End(Not run)
```

---

## sipREDCap *Default function to read from REDCap*

---

### Description

Default function to read from REDCap

### Usage

```
sipREDCap(key, ...)
```

## Arguments

| | |
|---|---|
| key | the api key of interest. The package provides this. |
| ... | Additional arguments passed to exportRecords. |

## Value

data.frame containing requested REDCap data.

## Examples

```
## Not run: data <- sipREDCap(keyring::key_get("rccola", "database_name", "project_name"))
```

# Index