

Package: datafaro (via r-universe)

November 12, 2024

Title Interact with the Datafaro API Seamlessly

Version 1.0.0

Description DataFaro offers access to recurrent data from a variety of sources. It abstracts concerns related to data frequency, format, and publication calendar. This package provides a convenient wrapper for interacting with the Datafaro API, making it easier for R users to harness the power of DataFaro.

License CC BY 4.0

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports pins, magrittr, httr2, jsonlite, glue, sodium, uuid, base64enc, cli, digest, getPass, lubridate, tibble, stringr,

Suggests knitr, rmarkdown, renviro

Depends R (>= 2.10), methods,

Remotes github::adatar-do/renviro

LazyData true

URL <https://datafaro.github.io/datafaroR/>

VignetteBuilder knitr

Config/pak/sysreqs make libicu-dev libsodium-dev libssl-dev

Repository <https://adatar-do.r-universe.dev>

RemoteUrl <https://github.com/datafaro/datafaroR>

RemoteRef HEAD

RemoteSha 2c98353139b5cf53c6a47b25608cafa28c099dce

Contents

as_datalight	2
auth	3

get_data	3
log_out	4
print.datalight	5
Index	6

as_datalight	<i>Convert a Datafaro's datalight list object to a datalight object</i>
--------------	---

Description

Convert a Datafaro's datalight list object to a datalight object

Usage

```
as_datalight(data)
```

Arguments

data A Datafaro's datalight list object

Value

a datalight object

Examples

```
## Not run:
.res <- list(
  data = mtcars,
  dictionary = list(),
  metadata = list(next_update = Sys.Date() + 30),
  cached = Sys.Date(),
  slug = "mtcars"
)
mi_tibble <- as_datalight(.res)
print(mi_tibble)

## End(Not run)
```

`auth`*Autenticar y obtener token de acceso*

Description

Esta función verifica si existe información de autenticación previa almacenada y válida. Si no encuentra información válida o el token ha expirado, solicita al usuario que se autentique nuevamente. Si la autenticación es exitosa, genera y almacena un nuevo token de acceso para solicitudes futuras. En caso de encontrar un token válido y no expirado, lo retorna para su uso inmediato.

Usage

```
auth(scope = "user")
```

Arguments

`scope` **character** Alcance de la autenticación. Por defecto es "user". Opciones válidas son "user" y "project". "user" estará disponible para todas las sesiones de R, mientras que "project" solo estará disponible para las sesiones de R en el proyecto actual.

Value

Retorna el token de acceso si la autenticación es exitosa o el token almacenado sigue siendo válido. En otros casos, guía al usuario para realizar una nueva autenticación.

Examples

```
## Not run:  
auth() # Ejecuta esta función para autenticarte o validar el token existente.  
  
## End(Not run)
```

`get_data`*Obtener la última actualización de un conjunto de datos de Datafaro*

Description

[Experimental]

Usage

```
get_data(code, .token = auth(), .cached = TRUE)
```

Arguments

code	character Código del conjunto de datos
. token	character Token de acceso a la API. Si no se ha guardado un token, se debe especificar el token manualmente.
. cached	logical Si se debe utilizar la caché. Por defecto es TRUE.

Value

Un datalight con la última actualización del conjunto de datos.

Examples

```
## Not run:
get_data("NC_XDC")

## End(Not run)
```

log_out

Cerrar sesión y eliminar el token de acceso

Description

Esta función elimina la información de autenticación almacenada para el alcance especificado, efectivamente cerrando la sesión del usuario. Esto es útil para garantizar que la información sensible no permanezca accesible en sesiones no autenticadas posteriormente.

Usage

```
log_out(scope = "user")
```

Arguments

scope	character Alcance de la autenticación para la que se desea cerrar sesión. Por defecto es "user". Las opciones válidas son "user" y "project". "user" afecta todas las sesiones de R, mientras que "project" solo afecta a la sesión de R en el proyecto actual.
-------	--

Value

No retorna valores. Invoca un mensaje que informa al usuario que la sesión ha sido cerrada correctamente.

Examples

```
## Not run:
log_out() # Ejecuta esta función para cerrar la sesión y eliminar el token de acceso.

## End(Not run)
```

print.datalight	<i>Print a datalight object</i>
-----------------	---------------------------------

Description

Print a datalight object

Usage

```
## S3 method for class 'datalight'  
print(x, ...)
```

Arguments

x	A Datafaro's datalight json object
...	Other arguments passed on to methods

Value

a print of the object. Invisible x

Examples

```
## Not run:  
.res <- jsonlite::toJSON(  
  list(  
    data = mtcars,  
    dictionary = list(),  
    metadata = list(next_update = Sys.Date() + 30),  
    cached = Sys.Date(),  
    slug = "mtcars"  
  )  
)  
mi_tibble <- as_datalight(.res)  
print(mi_tibble)  
  
## End(Not run)
```

Index

`as_datalight`, 2
`auth`, 3

`character`, 3, 4

`get_data`, 3

`log_out`, 4
`logical`, 4

`print.datalight`, 5