Package ‘rromeo’

May 16, 2019

Type Package

Title Access Publisher Copyright & Self-Archiving Policies via the 'SHERPA/RoMEO' API

Version 0.1.0

Description Fetches information from the 'SHERPA/RoMEO' API <http://www.sherpa.ac.uk/romeo/apimanual.php> which indexes policies of journal regarding the archival of scientific manuscripts before and/or after peer-review as well as formatted manuscripts.

License GPL-3

Encoding UTF-8

LazyData true

Imports httr, xml2

RoxygenNote 6.1.1

Suggests ggplot2, knitr, rmarkdown, testthat, vcr (>= 0.2.6), ISOcodes, mockery

Language en-US

VignetteBuilder knitr

URL https://github.com/ropensci/rromeo

BugReports https://github.com/ropensci/rromeo/issues

NeedsCompilation no

Author Matthias Grenié [aut, cre] (<https://orcid.org/0000-0002-4659-7522>), Hugo Gruson [aut] (<https://orcid.org/0000-0002-4094-1476>)

Maintainer Matthias Grenié <matthias.grenie@gmail.com>

Repository CRAN

Date/Publication 2019-05-16 16:30:03 UTC

R topics documented:

`check_key` *Check SHERPA/RoMEO API key*

Description

The key can be either specified in various ways see the Details section.

Usage

```
check_key(key = NULL)
```

Arguments

`key`	`[character(1)]` a character string containing the API key or `NULL` (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

if found the character string of the key, `NULL` otherwise

parse_generic *Generic parsing function*

Description

Generic parsing function

Usage

```
parse_generic(api_answer, ...)
```

Arguments

`api_answer`	`[httr::response()]` The API answer
`...`	Other options passed to parsing functions

Value

either results from `parse_journal()` or `parse_publisher()`

rr_api_version *Return SHERPA/RoMEO API version*

Description

This function queries SHERPA/RoMEO and returns the version of the API.

Usage

```
rr_api_version()
```

Examples

```
rr_api_version()
```

rr_auth *Store provided API key into Environment Variable*

Description

This function stores the provided API key as argument in to an environment variable `SHERPAROMEO_KEY` for further use by other `rromeo` functions.

Usage

```
rr_auth(key)
```

Arguments

key	`[character(1)]` A string giving the API key to save into the environment

Details

For more information regarding API keys, please refer to dedicated vignette with the following command `vignette("setting_up_api_key", package = "rromeo")`

Examples

```
## Not run:
rr_auth("Iq83AIL5bss")

## End(Not run)
```

rr_GET *rromeo internal GET function*

Description

rromeo internal GET function

Usage

```
rr_GET(...)
```

Arguments

...	additional parameter to `httr::GET`

rr_journal_find *Find if journals are available in SHERPA/RoMEO*

Description

Find if journals are available in SHERPA/RoMEO

Usage

```
rr_journal_find(name, qtype = c("exact", "contains", "starts"),
  key = NULL)
```

Arguments

name	[character(1+)] one or several strings to match the titles of the journals
qtype	[character(1)] in: * "exact" full title must be exactly to provided name, * "contains" the provided name must appear anywhere in the title of the journal, * "starts" the provided name must appear at the start of title of the journal.
key	[character(1)] a character string containing the API key or NULL (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: vignette("setting_up_api_key", package = "rromeo"). You can also use rr_auth that will use the provided key to store it as an environmental variable.

Value

Returns a data frame:

- title [character(1)]
  the name of the journal
- issn [character(1)]
  the ISSN of the journal

Examples

```
rr_journal_find(name = "Biostatistics", qtype = "contains")
```

rr_journal_issn *Retrieve journal policy using ISSN*

Description

Retrieve policy information from the SHERPA/RoMEO API using the ISSN of the journal

Usage

```
rr_journal_issn(issn, key = NULL)
```

Arguments

issn	[character(1+)] one or a vector of journal(s) ISSN(s)
key	[character(1)] a character string containing the API key or NULL (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

Returns a data.frame with complete information from journal:

- `title` [character(1)]
  the name of the journal
- `issn` [character(1)]
  the ISSN of the journal
- `romeocolour` [character(1)]
  the SHERPA/RoMEO colour of the journal
- `preprint` [character(1)]
  is the preprint (not reviewed) archivable?
- `postprint` [character(1)]
  is the postprint (reviewed but not formatted)?
- `pdf` [character(1)]
  is the publisher's version (reviewed and formatted)
- `pre_embargo` [character(1)]
  if applicable the embargo period before the author(s) can archive the preprint
- `post_embargo` [character(1)]
  if applicable the embargo period before the author(s) can archive the postprint

- pdf_embargo [character(1)]
  if applicable the embargo period before the author(s) can archive the publisher's version

Examples

```
rr_journal_issn(issn = "1947-6264")
rr_journal_issn(issn = c("1947-6264", "0030-1299"))
```

rr_journal_name *Retrieve journals policies by matching title*

Description

Note that SHERPARoMEO will not return more than 50 journals in a single query. The function will warn you if you are in this case.

Usage

```
rr_journal_name(name, qtype = c("exact", "contains", "starts"),
  key = NULL)
```

Arguments

name	[character(1+)] one or several strings to match the titles of the journals
qtype	[character(1)] in: * "exact" full title must be exactly to provided name, * "contains" the provided name must appear anywhere in the title of the journal, * "starts" the provided name must appear at the start of title of the journal.
key	[character(1)] a character string containing the API key or NULL (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: vignette("setting_up_api_key", package = "rromeo"). You can also use rr_auth that will use the provided key to store it as an environmental variable.

Value

Returns a data.frame with complete information from journal:

- `title` [character(1)]
  the name of the journal
- `issn` [character(1)]
  the ISSN of the journal
- `romeocolour` [character(1)]
  the SHERPA/RoMEO colour of the journal
- `preprint` [character(1)]
  is the preprint (not reviewed) archivable?
- `postprint` [character(1)]
  is the postprint (reviewed but not formatted)?
- `pdf` [character(1)]
  is the publisher's version (reviewed and formatted)
- `pre_embargo` [character(1)]
  if applicable the embargo period before the author(s) can archive the preprint
- `post_embargo` [character(1)]
  if applicable the embargo period before the author(s) can archive the postprint
- `pdf_embargo` [character(1)]
  if applicable the embargo period before the author(s) can archive the publisher's version

Examples

```
rr_journal_name(name = "Journal of Geology")
rr_journal_name(name = "Biogeography", qtype = "contains")
# You can also query multiple journals with exact titles in a single call
rr_journal_name(name = c("Journal of Biogeography", "PLoS ONE"),
                qtype = "exact")
```

`rr_publisher_all` *Get all Publisher Policies*

Description

Retrieve all data on publishers policies from SHERPA/RoMEO.

Usage

```
rr_publisher_all(key = NULL)
```

Arguments

`key`	`[character(1)]` a character string containing the API key or `NULL` (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

Returns a data frame with the following columns:

- `romeoid` `[integer(1)]`
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` `[character(1)]`
  the name of the publisher
- `alias` `[character(1)]`
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` `[character(1)]`
  a colour assigned by the database that reflects the default policies of the publisher
- `preprint` `[character(1)]`
  is the preprint (not reviewed) archivable?
- `postprint` `[character(1)]`
  is the postprint (reviewed but not formatted) archivable?
- `pdf` `[character(1)]`
  is the publisher's version (reviewed and formatted) archivable?

`rr_publisher_continent` *Get Publisher Policy by Publisher's Continent*

Description

Retrieve publisher's policy based on publisher's continent. This function does not work for unclassified or international publishers.

Usage

```
rr_publisher_continent(continent = c("Africa", "Antarctica", "Asia",
  "Australasia", "Caribbean", "Central America", "Europe", "North America",
  "Oceania", "South America"), key = NULL)
```

Arguments

`continent` [character(1+)]
one or a vector of strings in `c("Africa", "Antarctica", "Asia", "Australasia", "Carribean",`
the continent name to retrieve

`key` [character(1)]
a character string containing the API key or `NULL` (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

Returns a data frame with the following columns:

- `romeoid` [integer(1)]
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` [character(1)]
  the name of the publisher
- `alias` [character(1)]
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` [character(1)]
  a colour assigned by the database that reflects the default policies of the publisher
- `preprint` [character(1)]
  is the preprint (not reviewed) archivable?
- `postprint` [character(1)]
  is the postprint (reviewed but not formatted) archivable?
- `pdf` [character(1)]
  is the publisher's version (reviewed and formatted) archivable?

Examples

```
rr_publisher_continent(continent = "Caribbean")
rr_publisher_continent(continent = "Central America")
rr_publisher_continent(continent = c("Caribbean", "Central America"))
```

`rr_publisher_country` *Get Publisher Policy by Publisher's Country*

Description

Retrieve publisher's policy based on publisher's country. The code should be the ISO_3166-1_alpha-2 code of the country `https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2`.

Usage

```
rr_publisher_country(country, key = NULL)
```

Arguments

`country` `[character(1+)]`
one or a vector of ISO two-letter country code or `AA` for international publisher, `ZZ` for publisher of unknown countries and `__` for publishers without specified country (case insensitive).

`key` `[character(1)]`
a character string containing the API key or `NULL` (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

Returns a data frame with the following columns:

- `romeoid` `[integer(1)]`
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` `[character(1)]`
  the name of the publisher
- `alias` `[character(1)]`
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` `[character(1)]`
  a colour assigned by the database that reflects the default policies of the publisher
- `preprint` `[character(1)]`
  is the preprint (not reviewed) archivable?
- `postprint` `[character(1)]`
  is the postprint (reviewed but not formatted) archivable?
- `pdf` `[character(1)]`
  is the publisher's version (reviewed and formatted) archivable?

Examples

```
# Taiwan
rr_publisher_country("TW")
# Egypt
rr_publisher_country("EG")
rr_publisher_country(c("TW", "EG"))
```

---

rr_publisher_id *Get Publisher Policy from Publisher ID*

Description

Use SHERPA/RoMEO API to retrieve a specific publisher policies on manuscript archival

Usage

```
rr_publisher_id(id, key = NULL)
```

Arguments

id	[integer(1+)] one or a vector of SHERPA/RoMEO publisher's ID
key	[character(1)] a character string containing the API key or NULL (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: `vignette("setting_up_api_key", package = "rromeo")`. You can also use `rr_auth` that will use the provided key to store it as an environmental variable.

Value

Returns a data frame with the following columns:

- `romeoid` [integer(1)]
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` [character(1)]
  the name of the publisher
- `alias` [character(1)]
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` [character(1)]
  a colour assigned by the database that reflects the default policies of the publisher

- preprint [character(1)]
  is the preprint (not reviewed) archivable?
- postprint [character(1)]
  is the postprint (reviewed but not formatted) archivable?
- pdf [character(1)]
  is the publisher's version (reviewed and formatted) archivable?

Examples

```
rr_publisher_id(id = 55)
rr_publisher_id(id = c(55, 735))
```

rr_publisher_name *Get Publisher Policy by Publisher Name*

Description

Use SHERPA/RoMEO API to retrieve a specific publisher policies on manuscript archival based on matching the name of the publishers.

Usage

```
rr_publisher_name(name, qtype = c("all", "any", "exact"), key = NULL)
```

Arguments

name [character(1+)]
One or a vector of query string(s) to search publisher name

qtype [character(1)]
in c("all", "any", "exact") define the type of matching:

- all means that all strings in name must appear in any order or location
- any means that at least one of the strings in name must appear
- exact means that the name string must appear in the publisher's name or its alias.

key [character(1)]
a character string containing the API key or NULL (see Details section on how to specify it)

Details

There are several ways to provide your API key. The best way to know about them is to refer to the vignette about "Setting Up Your API key" accessible with the following command: vignette("setting_up_api_key", package = "rromeo"). You can also use rr_auth that will use the provided key to store it as an environmental variable.

Value

Returns a data frame with the following columns:

- `romeoid` [`integer(1)`]
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` [`character(1)`]
  the name of the publisher
- `alias` [`character(1)`]
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` [`character(1)`]
  a colour assigned by the database that reflects the default policies of the publisher
- `preprint` [`character(1)`]
  is the preprint (not reviewed) archivable?
- `postprint` [`character(1)`]
  is the postprint (reviewed but not formatted) archivable?
- `pdf` [`character(1)`]
  is the publisher's version (reviewed and formatted) archivable?

Examples

```
rr_publisher_name(name = "Optical Society", qtype = "all")
rr_publisher_name(name = "Swiss Chemistry", qtype = "any")
rr_publisher_name(name = "Swiss Chemistry", qtype = "exact")
```

rr_romeo_colour *Query publisher by RoMEO colour*

Description

SHERPA/RoMEO classifies publisher in different colours depending on their archiving policies.

- **green** publishers let authors archive preprint and postprint or publisher's version/PDF,
- **blue** publishers let authors archive postprint or publisher's version/PDF,
- **yellow** publishers let authors archive preprint,
- **white** publishers do not formally support archival.

Usage

```
rr_romeo_colour(romeo_colour = c("green", "blue", "yellow", "white"),
  key = NULL)
```

Arguments

`romeo_colour` [`character(1)`]
in `c("green", "blue", "yellow", "white")` the SHERPA/RoMEO colour to retrieve

`key` [`character(1)`]
a character string containing the API key or `NULL` (see Details section on how to specify it)

Details

For more details about the definitions of RoMEO colours check the FAQ section of SHERPA/RoMEO

Note that when using `rr_romeo_colour()` the API returns **all** the publishers in the selected category, so the results are generally bigger in size than specific functions like [`rr_journal_name()`] or [`rr_publisher_id()`]

Value

Returns a data frame with the following columns:

- `romeoid` [`integer(1)`]
  the internal index of the publisher in the SHERPA/RoMEO database
- `publisher` [`character(1)`]
  the name of the publisher
- `alias` [`character(1)`]
  if applicable an alternative name of the publisher or the name of the specific publishing branch
- `romeocolour` [`character(1)`]
  a colour assigned by the database that reflects the default policies of the publisher
- `preprint` [`character(1)`]
  is the preprint (not reviewed) archivable?
- `postprint` [`character(1)`]
  is the postprint (reviewed but not formatted) archivable?
- `pdf` [`character(1)`]
  is the publisher's version (reviewed and formatted) archivable?

Examples

```
rr_romeo_colour(romeo_colour = "green")
```

`rr_ua` *rromeo User Agent*

Description

rromeo User Agent

Usage

```
rr_ua()
```

Index