

Package ‘guardianapi’

February 3, 2019

Title Access 'The Guardian' Newspaper Open Data API

Version 0.1.0

Description Access to the 'Guardian' open API

<<https://open-platform.theguardian.com/>>, containing all articles published in 'The Guardian' from 1999 to the present, including article text, metadata, tags and contributor information. An API key and registration is required.

URL <https://docs.evanodell.com/guardianapi>

BugReports <https://github.com/evanodell/guardianapi/issues>

License MIT + file LICENSE

RoxygenNote 6.1.1

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

Imports dplyr, httr, jsonlite, tibble, rlang

Suggests testthat, knitr, rmarkdown, lubridate, ggplot2, covr, scales

VignetteBuilder knitr

NeedsCompilation no

Author Evan Odell [aut, cre] (<<https://orcid.org/0000-0003-1845-808X>>)

Maintainer Evan Odell <evanodell191@gmail.com>

Repository CRAN

Date/Publication 2019-02-03 15:33:14 UTC

R topics documented:

guardianapi	2
gu_api_key	2
gu_content	3
gu_editions	5
gu_items	6
gu_section	8
gu_tags	9

Index**11**

`guardianapi`*guardianapi*

Description

Access to the 'Guardian' open API <https://open-platform.theguardian.com/>, containing all articles, video and images published in the 'Guardian' from 1999 to the present. Users must **register** and use an API key, which can be saved with the `gu_api_key()` function, or as the GU_API_KEY environmental variable. Free users can make up to 5,000 calls per day and 12 calls per second, and access all article text and associated metadata. Images and video require a commercial subscription.

`gu_api_key`*API Key*

Description

A function to assign or re-assign the API key. Register for an API key on the **Guardian Open Platform** site.

By default, `guardianapi` will look for the GU_API_KEY environmental variable when the package is loaded. If found the key is stored in the session option `gu.API.key`. This function can be used to set the key for a single session. To avoid having to use this function, use the `.Renviro` file to store you key as GU_API_KEY.

Usage

```
gu_api_key(check_env = FALSE)
```

Arguments

`check_env` If TRUE, will check the environment variable GU_API_KEY first before asking for user input. If found, assigns the API key to the `gu.API.key` variable.

gu_content

*Content***Description**

Query and return all available content in the API.

See the [API docs](#) for full details on the query options available for this content endpoint.

Usage

```
gu_content(query = NULL, show_fields = "all", show_tags = "all",
  tag = NULL, from_date = NULL, to_date = NULL,
  use_date = "published", ..., verbose = TRUE, tidy = TRUE,
  tidy_style = "snake_case")
```

Arguments

query	A string, containing the search query. Defaults to NULL, which returns all available content subject to other parameters. Supports AND, OR and NOT operators, and exact phrase queries using double quotes. E.g. '"football" OR "politics"'.
show_fields	A string or character vector of fields to include in the returned data. Defaults to "all". See details for a list of options.
show_tags	A string or character vector of tags to include in the returned data. Defaults to "all". See details for a list of options.
tag	A string or character vector of tags to filter the returned data. Defaults to NULL.
from_date	Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to NULL.
to_date	Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to NULL.
use_date	The date type to use for the from_date and to_date parameters. One of "published", "first-publication", "newspaper-edition" or "last-modified". Defaults to "published".
...	Use to pass any other parameters to the API. See the docs for a full list of options.
verbose	Prints messages to console. Defaults to TRUE.
tidy	Convert variable names to snake_case, remove some "<NA>" strings. Defaults to TRUE.
tidy_style	Style to variable names with.

Value

A tibble.

Fields options

The following are the options for the `show_fields` parameter:

- "all" Includes all the fields (default)
- "trailText"
- "headline"
- "showInRelatedContent" Whether this content can appear in automatically generated Related Content
- "body"
- "lastModified"
- "hasStoryPackage" Has related content selected by editors
- "score" A relevance score based on the search query used
- "standfirst"
- "shortUrl"
- "thumbnail"
- "wordcount"
- "commentable"
- "isPremoderated" Comments will be checked by a moderator prior to publication if true.
- "allowUgc" May have associated User Generated Content. This typically means the content has an associated Guardian Witness assignment which can be accessed by querying "show-references=witness-assignment", using the query parameter.
- "byline"
- "publication"
- "internalPageCode"
- "productionOffice"
- "shouldHideAdverts" Adverts will not be displayed if true
- "liveBloggingNow" Content is currently live blogged if true
- "commentCloseDate" The date the comments have been closed
- "starRating"

#? The following are the options for the `show_tags` parameter

- "blog"
- "contributor"
- "keyword"
- "newspaper-book"
- "newspaper-book-section"
- "publication"
- "series"
- "tone"
- "type"
- "all": The default option.

Examples

```
## Not run:
x <- gu_content(query = "films")

y <- gu_content(
  query = "relationships",
  from_date = "2018-11-30", to_date = "2018-12-30"
)

## End(Not run)
```

gu_editions	<i>Editions</i>
-------------	-----------------

Description

The different main pages of the Guardian. As of January 2019 they are the United Kingdom ("uk"), the United States ("us"), Australia ("au") and an International ("international") front page.

Usage

```
gu_editions(query = NULL, ..., verbose = TRUE, tidy = TRUE,
  tidy_style = "snake_case")
```

Arguments

query	A string, which will return editions based on that string. Defaults to NULL and returns all editions. Strings are not case sensitive.
...	Pass additional options to API. There are no additional options as of this writing. See the endpoint docs
verbose	Prints messages to console. Defaults to TRUE.
tidy	Convert variable names to snake_case, remove some "<NA>" strings. Defaults to TRUE.
tidy_style	Style to variable names with.

Value

A tibble with details of the given edition.

Examples

```
## Not run:
uk <- gu_editions(query = "uk")

## End(Not run)
```

gu_items

*Items***Description**

Query and return one or more API items.

See the [API docs](#) for full details on the query options available for this endpoint.

Usage

```
gu_items(query = NULL, show_fields = "all", show_tags = "all",
         tag = NULL, from_date = NULL, to_date = NULL,
         use_date = "published", ..., verbose = TRUE, tidy = TRUE,
         tidy_style = "snake_case")
```

Arguments

query	A string, containing the search query, either the URL of a single item or all items listed under a given profile, tag, etc. For example, to return all articles by a given contributor, use "profile/{contributorname}", e.g. "profile/brianlogan".
show_fields	A string or character vector of fields to include in the returned data. Defaults to "all". See details for a list of options.
show_tags	A string or character vector of tags to include in the returned data. Defaults to "all". See details for a list of options.
tag	A string or character vector of tags to filter the returned data. Defaults to NULL.
from_date	Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to NULL.
to_date	Accepts character values in 'YYYY-MM-DD' format, and objects of class Date, POSIXt, POSIXct, POSIXlt or anything else that can be coerced to a date with as.Date(). Defaults to NULL.
use_date	The date type to use for the from_date and to_date parameters. One of "published", "first-publication", "newspaper-edition" or "last-modified". Defaults to "published".
...	Use to pass any other parameters to the API. See the item docs for a full list of options, including those not included here.
verbose	Prints messages to console. Defaults to TRUE.
tidy	Convert variable names to snake_case, remove some "<NA>" strings. Defaults to TRUE.
tidy_style	Style to variable names with.

Value

A tibble.

Fields options

The following are the options for the show_fields parameter:

- "all" Includes all the fields (default)
- "trailText"
- "headline"
- "showInRelatedContent" Whether this content can appear in automatically generated Related Content
- "body"
- "lastModified"
- "hasStoryPackage" Has related content selected by editors
- "score" A relevance score based on the search query used
- "standfirst"
- "shortUrl"
- "thumbnail"
- "wordcount"
- "commentable"
- "isPremoderated" Comments will be checked by a moderator prior to publication if true.
- "allowUgc" May have associated User Generated Content. This typically means the content has an associated Guardian Witness assignment which can be accessed by querying "show-references=witness-assignment", using the query parameter.
- "byline"
- "publication"
- "internalPageCode"
- "productionOffice"
- "shouldHideAdverts" Adverts will not be displayed if true
- "liveBloggingNow" Content is currently live blogged if true
- "commentCloseDate" The date the comments have been closed
- "starRating"

The following are the options for the show_tags parameter

- "blog"
- "contributor"
- "keyword"
- "newspaper-book"
- "newspaper-book-section"
- "publication"
- "series"
- "tone"
- "type"
- "all": The default option.

Examples

```
## Not run:
x <- gu_content(query = "films")

## End(Not run)
```

gu_section

Sections

Description

Returns details on the sections and subsections used to organise content.

See the [API docs](#) for full details on the query options available for the sections endpoint.

Usage

```
gu_section(query = NULL, ..., verbose = TRUE, tidy = TRUE,
           tidy_style = "snake_case")
```

Arguments

query	A string, containing the search query. Defaults to NULL, which returns all available sections subject to other parameters. Supports AND, OR and NOT operators, and exact phrase queries using double quotes. E.g. '"football" OR "politics"'. Also accepts a character vector of section names and returns those sections.
...	Use to pass any other parameters to the API. See the docs for a full list of options.
verbose	Prints messages to console. Defaults to TRUE.
tidy	Convert variable names to snake_case, remove some "<NA>" strings. Defaults to TRUE.
tidy_style	Style to variable names with.

Examples

```
## Not run:
business <- gu_section(query = "business")

foot_pol <- gu_section(query = c("politics", "business", "football"))

## End(Not run)
```

gu_tags

*Tags***Description**

All the tags used on the Guardian website. See the [API docs](#) on this endpoint for more details.

Usage

```
gu_tags(query = NULL, tag_type = NULL, section = NULL,
        references = NULL, reference_type = NULL, show_references = "all",
        ..., verbose = TRUE, tidy = TRUE, tidy_style = "snake_case")
```

Arguments

query	A string, which will return all tags containing that string.
tag_type	One of "keyword", "series", "contributor", "tone", "type" or "blog". Defaults to NULL and does not filter by tag type.
section	Return only tags of a given section.
references	Return only tags with those references
reference_type	Return only tags with those reference types.
show_references	Show associated reference data such as ISBNs. Defaults to "all" and shows all available references. Accepts character vectors of one or more references (see details for options).
...	Use to pass any other parameters to the API. See the docs for a full list of options.
verbose	Prints messages to console. Defaults to TRUE.
tidy	Convert variable names to snake_case, remove some "<NA>" strings. Defaults to TRUE.
tidy_style	Style to variable names with.

Value

A tibble with details on tags.

References options

The following are the options for the show_references parameter:

- "all" Includes all the fields (default)
- "author"
- "bisac-prefix"
- "esa-cricket-match"
- "esa-football-match"

- "esa-football-team"
- "esa-football-tournament"
- "isbn"
- "imdb"
- "musicbrainz"
- "musicbrainzgenre"
- "opta-cricket-match"
- "opta-football-match"
- "opta-football-team"
- "opta-football-tournament"
- "pa-football-competition"
- "pa-football-match"
- "pa-football-team"
- "r1-film"
- "reuters-index-ric"
- "reuters-stock-ric"
- "witness-assignment"

Examples

```
## Not run:  
# Return all tags containing "apple"  
apple1 <- gu_tags(query = "apple")  
  
# Return all tags containing "apple" in the technology section  
apple2 <- gu_tags(query = "apple", section = "technology")  
  
# Return all contributor tags in the life and style section  
tag_sec_type <- gu_tags(section = "lifeandstyle", tag_type = "contributor")  
  
## End(Not run)
```

Index

[gu_api_key](#), [2](#)
[gu_api_key\(\)](#), [2](#)
[gu_content](#), [3](#)
[gu_editions](#), [5](#)
[gu_items](#), [6](#)
[gu_section](#), [8](#)
[gu_tags](#), [9](#)
[guardianapi](#), [2](#)
[guardianapi-package \(guardianapi\)](#), [2](#)