Package ‘rsunlight’

May 15, 2018

Title  Interface to ‘Sunlight’ Foundation ‘APIs’

Description  Interface to ‘APIs’ for US government data previously under the ‘Sunlight’ Foundation, now under ‘ProPublica’ (<https://www.propublica.org/datastore//apis>), and Open States (<http://docs.openstates.org/en/latest/api/>). Functions are provided to interact with each of the routes in each API.

Version  0.7.0

License  MIT + file LICENSE

URL  https://github.com/ropengov/rsunlight

BugReports  https://github.com/ropengov/rsunlight/issues

LazyData  true

Imports  methods, utils, crul, plyr, jsonlite, stringr, tibble, data.table

Suggests  testthat

Enhances  ggplot2

RoxygenNote  6.0.1

NeedsCompilation  no

Author  Scott Chamberlain [aut, cre], Thomas J. Leeper [ctb]

Maintainer  Scott Chamberlain <myrmecocystus@gmail.com>

Repository  CRAN

Date/Publication  2018-05-15 06:29:45 UTC

R topics documented:

rsunlight-package ......................................................... 2
cg_bill ................................................................. 3
cg_bills ................................................................. 4
cg_bills_recent ......................................................... 4
cg_bills_recent_member ............................................... 5
Index

rsunlight-package  Methods for retrieving data from APIs that were previously under Sunlight Labs

Description

The various APIs included here are now managed by ProPublica and OpenStates organizations.

Details

You need API keys for Propublica APIs. Get them at https://www.propublica.org/datastore/apis

You need an API key for the OpenStates API. Get it at https://openstates.org/api/register/

We set up the functions so that you can use either env vars, or R options. For env vars, put an entry in your .Renviron file with the names PROPUBLICA_API_KEY and OPEN_STATES_KEY.

Currently we have functions to interface with the following Propublica API and OpenStates API:

- ProPublica Congress API (cg)
• Open States API (os)

NOTES:
• Capitol Words API is now defunct
• We will support ProPublica campaign finance API in the future

Author(s)
Scott Chamberlain <myrmecocystus@gmail.com>
Thomas J. Leeper <thosjleeper@gmail.com>

cg_bill

Get a specific bill

Description
Get a specific bill

Usage
cg_bill(congress, id, key = NULL, as = "table", ...)

Arguments

congress (character) congress number, e.g. 110
id (character) a bill slug, for example hr4881 - these can be found in the recent bill response.
key your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
as (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
... Optional additional curl options (debugging tools mostly) passed on to HttpClient

Value
some nasty parsing, so just list of lists for now

Examples
### Not run:
cg_bill(115, "hr21")

### End(Not run)
### cg_bills

**Search bills**

**Description**
Search bills

**Usage**
```
cg_bills(query, sort = NULL, dir = NULL, key = NULL, as = "table", ...)
```

**Arguments**
- `query` (character) keyword or phrase. required
- `sort` (character) ".score" or "date" (default)
- `dir` (character) asc or desc (default)
- `key` (character) your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- `as` (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- `...` Optional additional curl options (debugging tools mostly) passed on to HttpClient

**Value**
some nasty parsing, so just list of lists for now

**Examples**
```
## Not run:
cg_bills(query = "megahertz")
cg_bills(query = "megahertz", per_page = 3)
## End(Not run)
```

### cg_bills_recent

**Get recent bills**

**Description**
Get recent bills

**Usage**
```
cg_bills_recent(congress, chamber, type, key = NULL, as = "table", ...)
```
cg_bills_recent_member

Arguments

- **congress**: The number of the Congress this update took place during.
- **chamber**: The chamber this update took place in. 'house' or 'senate'
- **type**: Type of bill. introduced, updated, active, passed, enacted or vetoed
- **key**: your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- **as**: (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

Value

some nasty parsing, so just list of lists for now

Examples

```r
## Not run:
cg_bills_recent(115, "house", "introduced")
## End(Not run)
```

---

cg_bills_recent_member

*Get recent bills by member id*

Description

Get recent bills by member id

Usage

```r
cg_bills_recent_member(id, type, key = NULL, as = "table", ...)
```

Arguments

- **id**: (character) The ID of the member to retrieve; it is assigned by the Biographical Directory of the United States Congress [(http://bioguide.congress.gov/biosearch/biosearch.asp)](http://bioguide.congress.gov/biosearch/biosearch.asp) or can be retrieved from a cg_members() request
- **type**: Type of bill. introduced or updated
- **key**: your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- **as**: (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

... Optional additional curl options (debugging tools mostly) passed on to HttpClient
**cg_bills_recent_subject**

*Get recent bills by subject*

**Description**

Get recent bills by subject

**Usage**

```r
cg_bills_recent_subject(subject, key = NULL, as = "table", ...)
```

**Arguments**

- `subject` (character) A slug version of a legislative subject, displayed as `url_name` in subject responses.
- `key` (character) Your SunlightLabs API key; loads from environment variable from `.Renviron` or from an option from `.Rprofile`
- `as` (character) One of `table` (default), `list`, or `response` (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

**Value**

Some nasty parsing, so just list of lists for now

**Examples**

```r
## Not run:
cg_bills_recent_subject("L000287", "introduced")
## End(Not run)
```
cg_bills_upcoming  

Search for upcoming bills

Description

Search for upcoming bills

Usage

cg_bills_upcoming(chamber, key = NULL, as = "table", ...)

Arguments

chamber  The chamber which has scheduled this bill. house or senate
key  your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
as  (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
...  Optional additional curl options (debugging tools mostly) passed on to **HttpClient**

Examples

```r
## Not run:
cg_bills_upcoming("house")
cg_bills_upcoming("senate")
## End(Not run)
```

cg_bill_amendments  

Search for data on amendments for a specific bill

Description

Search for data on amendments for a specific bill

Usage

cg_bill_amendments(congress, id, key = NULL, as = "table", ...)

Arguments

congress congress id (numeric.integer)
id the bill id (character)
key your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
as (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

Examples

## Not run:
cg_bill_cosponsors(115, 'hr1628')

## End(Not run)

cg_bill_cosponsors Get cosponsors for a specific bill

Description

Get cosponsors for a specific bill

Usage

cg_bill_cosponsors(congress, id, key = NULL, as = "table", ...)

Arguments

congress congress id (numeric.integer)
id the bill id (character)
key your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
as (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

Examples

## Not run:
cg_bill_cosponsors(114, 'hr4249')

## End(Not run)
**cg_bill_related_bills**  
*Get related bills for a specific bill*

**Description**

Get related bills for a specific bill

**Usage**

`cg_bill_related_bills(congress, id, key = NULL, as = "table", ...)`

**Arguments**

- `congress`: congress id (numeric, integer)
- `id`: the bill id (character)
- `key`: your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- `as`: (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- `...`: Optional additional curl options (debugging tools mostly) passed on to `HttpClient`

**Examples**

```r
## Not run:
cg_bill_related_bills(115, 'hr3219')

## End(Not run)
```

---

**cg_bill_subjects**  
*Get Library of Congress-assigned subjects about a particular bill*

**Description**

Get Library of Congress-assigned subjects about a particular bill

**Usage**

`cg_bill_subjects(congress, id, key = NULL, as = "table", ...)`
Arguments

- **congress**: congress id (numeric, integer)
- **id**: the bill id (character)
- **key**: your SunlightLabs API key; loads from environment variable from `.Renviron` or from an option from `.Rprofile` (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- **as**: (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

Examples

```r
## Not run:
cg_bill_subjects_search('climate')
```

---

**cg_bill_subjects_search**

*Search for bill subjects that contain a specified term*

Description

Search for bill subjects that contain a specified term

Usage

```r
cg_bill_subjects_search(query, key = NULL, as = "table", ...)
```

Arguments

- **query**: (character) a word or phrase to search
- **key**: your SunlightLabs API key; loads from environment variable from `.Renviron` or from an option from `.Rprofile` (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- **as**: (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.

Examples

```r
## Not run:
cg_bill_subjects_search('climate')
```

## End(Not run)
**cg_state_party_countes**

*state party counts*

**Description**

state party counts

**Usage**

```r
cg_state_party_countes(key = NULL, as = "table", ...)
```

**Arguments**

- `key` (character) your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- `as` (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- `...` Optional additional curl options (debugging tools mostly) passed on to HttpClient

**Examples**

```r
## Not run:
cg_state_party_countes()
## End(Not run)
```

**committees**

*committees methods*

**Description**

committees methods

**Usage**

```r
cg_committees(congress, chamber, key = NULL, as = "table", ...)

cg_committee(congress, chamber, id, sub_id = NULL, key = NULL, as = "table", ...)

cg_committee_hearings(congress, chamber = NULL, id = NULL, key = NULL, as = "table", ...)
```
committees

cg_committee_comms(congress, key = NULL, as = "table", ...)

cg_committees_comms_category(congress, category, key = NULL, as = "table", ...)

cg_committees_comms_date(date, key = NULL, as = "table", ...)

cg_committees_comms_chamber(congress, chamber, key = NULL, as = "table", ...)

Arguments

congress (character) The number of the Congress this update took place during.
chamber (character) The chamber this update took place in. 'house' or 'senate'.
key your ProPublica API key; pass in or loads from environment variable stored as
PROPUBLICA_API_KEY in either your .Renviron, or similar file located in your
home directory
as (character) IGNORED FOR NOW
... optional curl options passed on to curl::HttpClient. See curl::curl_options()
id (character) a committee id
sub_id (character) a sub-committee id
category (character) one of ec, pm, or pom
date (character) a date, of the form YYYY-MM-DD

Value

various things for now, since return objects vary quite a bit among the different votes routes

Examples

## Not run:
cg_committees(115, "senate")
cg_committee(115, "senate", "SSAF")
cg_committee(115, "senate", "HSAS", sub_id = "HSAS28")
cg_committee_hearings(115)
cg_committee_hearings(115, "house", "HSRU")
cg_committee_comms(115)
cg_committees_comms_category(115, "pm")
cg_committees_comms_date("2018-03-21")
cg_committees_comms_chamber(115, "house")

## End(Not run)
floor_actions

floor_actions methods

Description

floor actions methods

Usage

```r
cg_floor_actions_recent(congress, chamber, key = NULL, as = "table", ...)
cg_floor_actions_date(chamber, year, month, day, key = NULL, as = "table", ...)
```

Arguments

- **congress** (character) The number of the Congress this update took place during.
- **chamber** (character) The chamber this update took place in. 'house' or 'senate'.
- **key** your ProPublica API key; pass in or loads from environment variable stored as PROPUBLICA_API_KEY in either your .Renviron, or similar file locatd in your home directory
- **as** (character) IGNORED FOR NOW
- **...** optional curl options passed on to `crl::HttpClient`. See `crl::crl_options()`
- **year** (integer) year of the form YYYY
- **month** (integer) month of the form MM
- **day** (integer) day of the form DD

Value

various things for now, since return objects vary quite a bit among the different votes routes

Examples

```r
## Not run:
cg_floor_actions_recent(115, "house")
cg_floor_actions_date("senate", 2017, 5, 2)

## End(Not run)
```
members  Members functions

Description
Members functions

Usage

cg_members(congress, chamber, key = NULL, as = "table", ...)

cg_member(member, key = NULL, as = "table", ...)

cg_members_new(key = NULL, as = "table", ...)

cg_members_state_district(chamber, state, district = NULL, key = NULL,
as = "table", ...)

cg_members_leaving(congress, chamber, key = NULL, as = "table", ...)

cg_members_votes(member, key = NULL, as = "table", ...)

cg_members_compare_votes(member1, member2, congress, chamber, key = NULL,
as = "table", ...)

cg_members_compare_bill_sponsors(member1, member2, congress, chamber,
key = NULL, as = "table", ...)

cg_members_bill_cosponsors(member, type, key = NULL, as = "table", ...)

Arguments

congress  The number of the Congress this update took place during.
chamber  The chamber this update took place in. ‘house’ or ‘senate’.
key  your ProPublica API key; pass in or loads from environment variable stored as
PROPUBLICA_API_KEY in either your .Renviron, or similar file located in your
home directory
as  (character) IGNORED FOR NOW
...  optional curl options passed on to crul::HttpClient. See curl::curl_options()
member, member1, member2  A member ID
state  A two character state code
district  House of Representatives district number (House requests only)
type  Type of bill. cosponsored or withdrawn
nominations

Value

various things for now, since return objects vary quite a bit among the different members routes

Examples

```r
# Not run:
cg_members(congress = 115, chamber = "senate")
cg_member(member = 'K000388')
cg_members_new()
cg_members_state_district('senate', 'RI')
cg_members_leaving(congress = 115, chamber = "house")
cg_members_votes('K000388')
cg_members_compare_votes("G000575", "D000624", 114, "house")
cg_members_compare_bill_sponsors("G000575", "D000624", 114, "house")
cg_members_bill_cosponsors("B001260", "cosponsored")

# End(Not run)
```

---

nominations methods

Description

nominations methods

Usage

```r
cg_nomination(congress, id, key = NULL, as = "table", ...)

cg_nominations_category(congress, type, key = NULL, as = "table", ...)

cg_nominations_state(congress, state, key = NULL, as = "table", ...)
```

Arguments

- **congress** (integer/numeric) the Congress in which this nomination was presented
- **id** alphanumeric ID beginning with PN - for example, "PN675"
- **key** your SunlightLabs API key: loads from environment variable from .Renviron or from an option from .Rprofile
- **as** (character) One of table (default), list, or response (httr response object). When table is requested, the default, a table is not always returned, but is returned almost always. You can then make a table yourself if you like.
- **type** (character) received, updated, confirmed, or withdrawn
- **state** (character) state Two-letter state abbreviation
os_billlookup

Details

- cg_nomination: Get a specific nomination
- cg_nominations_category: Get recent nominations by category
- cg_nominations_state: Get nominees by state

Examples

```r
## Not run:
cg_nomination(115, "PN40")
cg_nominations_category(115, "confirmed")
cg_nominations_state(115, "VA")
## End(Not run)
```

os_billlookup  

Look up bills on OpenStates.

Description

Look up bills on OpenStates.

Usage

```r
os_billlookup(state = NULL, session = NULL, bill_id = NULL, fields = NULL, per_page = NULL, page = NULL, as = "table", key = NULL, ...)`n```

Arguments

- **state**  
  state two-letter abbreviation (character), required
- **session**  
  session of congress (integer), e.g., 2009-2010 = 20092010, required
- **bill_id**  
  One or more identification numbers of bills (character), required
- **fields**  
  You can request specific fields by supplying a vector of fields names. Many fields are not returned unless requested. If you don’t supply a fields parameter, you will get the most commonly used subset of fields only. To save on bandwidth, parsing time, and confusion, it’s recommended to always specify which fields you will be using.
- **per_page**  
  Number of records to return. Default: 20. Max: 50.
- **page**  
  Page to return. Default: 1. You can use this in combination with the per_page parameter to get more than the default or max number of results per page.
- **as**  
  (character) One of table (default), list, or response (curl response object)
- **key**  
  your SunlightLabs API key; loads from .Rprofile
- **...**  
  Curl options passed on to `curl::HttpClient`
os_billsearch

Value

List of output fields.

Examples

## Not run:
os_billlookup(state='ca', session=20092010, bill_id='AB 667')
os_billlookup(state='ca', session=20092010, bill_id='AB 667', per_page=1)
os_billlookup(state='ca', session=20092010, bill_id='AB 667', per_page=1, fields='id')
os_billlookup(state='ca', session=20092010, bill_id='AB 667', per_page=3, fields=c('id','title'))
os_billlookup(state='ca', session=20092010, bill_id='SB 425')
os_billlookup(state='ca', session=20092010, bill_id=c('AB 667','SB 425'))
os_billlookup(state='ca', session=20092010, bill_id='AB 667', verbose = TRUE)

## End(Not run)

---

os_billsearch  Search OpenStates bills.

---

Description

Search OpenStates bills.

Usage

```r
os_billsearch(terms = NULL, state = NULL, window = NULL,
chamber = "upper", sponsor_id = NULL, updated_since = NULL,
subject = NULL, type = NULL, search_window = NULL, sort = NULL,
page = NULL, per_page = NULL, fields = NULL, as = "table",
key = NULL, ...)
```

Arguments

terms search terms bill search (character)
state state two-letter abbreviation (character)
window a string representing what time period to search across. Pass 'session’ to search bills from the state’s current or most recent legislative session, ‘term’ to search the current or most recent term, ‘all’ to search as far back as the Open State Project has data for, or supply 'session:SESSION_NAME’ or 'term:TERM_NAME’ (e.g. ‘session:2009’ or 'term:2009-2010’) to search a specific session or term.
chamber one of 'upper’ or 'lower’ (character)
sponsor_id only return bills sponsored by the legislator with the given id (corresponds to leg_id)
updated_since only return bills that have been updated since a given date, YYYY-MM-DD format
subject  filter by bills that are about a given subject. If multiple subject parameters are supplied then only bills that match all of them will be returned. See list of subjects

type  (character) Only bills of a given type (e.g. 'bill', 'resolution', etc.)

search_window  By default all bills are searched, but if a time window is desired the following options can be passed to search_window: all (Default, include all sessions.), term (Only bills from sessions within the current term.), session (Only bills from the current session.), session:2009 (Only bills from the session named 2009.), term:2009-2011 (Only bills from the sessions in the 2009-2011 session.)

sort  (character) One of 'first' (default), 'last', 'signed', 'passed_lower', 'passed_upper', 'updated_at', or 'created_at'.

page  Page to return. Default: 1. You can use this in combination with the per_page parameter to get more than the default or max number of results per page.

per_page  Number of records to return. Default: 20. Max: 50.

fields  You can request specific fields by supplying a vector of fields names. Many fields are not returned unless requested. If you don’t supply a fields parameter, you will get the most commonly used subset of fields only. To save on bandwidth, parsing time, and confusion, it’s recommended to always specify which fields you will be using.

as  (character) One of table (default), list, or response (crul response object)

key  your SunlightLabs API key; loads from .Rprofile

...  Curl options passed on to curl::HttpClient

Value

a data.frame of bills.

Examples

```r
# Not run:
os_billsearch(terms = 'agriculture', state = 'tx')
os_billsearch(terms = 'agriculture', state = 'tx', chamber = 'upper')
os_billsearch(terms = 'taxi', state = 'dc')
os_billsearch(terms = 'taxi', state = 'dc', per_page=3)
os_billsearch(terms = 'taxi', state = 'dc', per_page=3, sort='created_at')
os_billsearch(terms = 'taxi', state = 'dc', type='resolution')

# Search window
os_billsearch(terms = 'climate change', search_window='term')
os_billsearch(terms = 'climate change', search_window='term:2009-2011')
os_billsearch(terms = 'climate change', search_window='session')
os_billsearch(terms = 'climate change', search_window='session:2009')
os_billsearch(terms = 'agriculture', state = 'tx', per_page=2)
os_billsearch(terms = 'agriculture', state = 'tx', per_page=2, page=2)
os_billsearch(terms = 'agriculture', state = 'tx', fields=c('id','created_at'), per_page=10)

# Pass in more than one value for some parameters
```

os_committee

os_billsearch(terms = 'agriculture', state = c('tx', 'or'))

## End(Not run)

---

os_committee  

**Lookup a Committee by ID**

**Description**  
Lookup a Committee by ID

**Usage**  

```r
os_committee(id, as = "table", key = NULL, ...)
```

**Arguments**  

- **id**: committee id (character)
- **as**: (character) One of table (default), list, or response
- **key**: your SunlightLabs API key: loads from .Rprofile
- **...**: Curl options passed on to `crl::HttpClient`

**Value**  
List of output fields.

**Examples**  

```r
## Not run:
os_committee(id = 'TXC000103')

## End(Not run)
```

---

os_committees  

**Search OpenStates committees**

**Description**  
Search OpenStates committees

**Usage**  

```r
os_committees(state = NULL, chamber = "upper", committee = NULL,
subcommittee = NULL, as = "table", key = NULL, ...)
```
Arguments

state  state two-letter abbreviation (character)
chamber one of 'upper' or 'lower' (character)
committee a committee name (character)
subcommittee a subcommittee name (character)
as (character) One of table (default), list, or response (crl::HttpClient)
key your SunlightLabs API key; loads from .Rprofile
... Curl options passed on to crul::HttpClient

Details

this route appears to not support pagination, sorting or selecting fields

Value

a data.frame of bills.

Examples

```r
## Not run:
os_committees(state = 'tx')
os_committees(state = 'tx', chamber = 'upper')
os_committees(state = 'dc')

## End(Not run)
```

---

**os_district**  
*Get OpenStates districts boundary by boundary id*

Description

Get OpenStates districts boundary by boundary id

Usage

```r
os_district(id, as = "table", key = NULL, ...)
```

Arguments

id a boundary id used in District Boundary Lookup (character)
as (character) One of table (default), list, or response (crl::HttpClient)
key your SunlightLabs API key; loads from .Rprofile
... Curl options passed on to crul::HttpClient
Value

a data.frame of bills.

Examples

```r
## Not run:
os_district(id = 'ocd-division/country:us/state:tx/sldl:100')

## End(Not run)
```

---

### os_districts Search OpenStates districts

**Description**

Search OpenStates districts

**Usage**

```r
os_districts(state, chamber = NULL, abbr = NULL, boundary_id = NULL,
             id = NULL, legislators = NULL, name = NULL, num_seats = NULL,
             as = "table", key = NULL, ...)
```

**Arguments**

- **state**
  - state two-letter abbreviation (character). required
- **chamber**
  - Whether this district belongs to the upper or lower chamber. (character)
- **abbr**
  - State abbreviation. (character)
- **boundary_id**
  - boundary_id used in District Boundary Lookup (character)
- **id**
  - A unique ID for this district (separate from boundary_id).
- **legislators**
  - List of legislators that serve in this district. (may be more than one if num_seats > 1)
- **name**
  - Name of the district (e.g. '14', '33A', 'Fifth Suffolk')
- **num_seats**
  - Number of legislators that are elected to this seat. Generally one, but will be 2 or more if the seat is a multi-member district.
- **as**
  - (character) One of table (default), list, or response (crl::HttpClient response object)
- **key**
  - your SunlightLabs API key; loads from .Rprofile
- **...**
  - Curl options passed on to `crl::HttpClient`

**Details**

this route appears to not support pagination, sorting or selecting fields
Value

a data.frame of bills.

Examples

```
## Not run:
os_districts(state = 'tx')
os_districts(state = 'tx', chamber = 'upper')
os_districts(state = 'dc')
os_districts(state = 'dc', per_page=3)
os_districts(state = 'dc', per_page=3, sort='created_at')

## End(Not run)
```

---

**os_legislator**  
*Lookup a Legislator by ID*

Description

Lookup a Legislator by ID

Usage

```
os_legislator(id, as = "table", key = NULL, ...)
```

Arguments

- **id**  
  legislator id (character)
- **as**  
  (character) One of table (default), list, or response
- **key**  
  your SunlightLabs API key; loads from .Rprofile
- **...**  
  Curl options passed on to `crl::HttpClient`

Value

List of output fields.

Examples

```
## Not run:
os_legislator(id = 'DCL000012')

## End(Not run)
```
Description

Lookup Legislators geospatially

Usage

```
os_legislatorgeo(lat, long, as = "table", key = NULL, ...)
```

Arguments

- **lat**: a latitude value (numeric,integer)
- **long**: a longitude value (numeric,integer)
- **as**: (character) One of table (default), list, or response
- **key**: your SunlightLabs API key; loads from .Rprofile
- **...**: Curl options passed on to `cru::HttpClient`

Value

various, depending on as parameter

Examples

```R
## Not run:
os_legislatorgeo(lat = 35.79, long = -78.78)
## End(Not run)
```

Description

Search Legislators on OpenStates.

Usage

```
os_legislatorsearch(state = NULL, first_name = NULL, last_name = NULL, chamber = NULL, active = NULL, term = NULL, district = NULL, party = NULL, fields = NULL, as = "table", key = NULL, ...)
```
Arguments

state  state two-letter abbreviation (character)
first_name  first name of legislator (character)
last_name  last name of legislator (character)
chamber  one of 'upper' or 'lower' (character)
active  TRUE or FALSE (character)
term  filter by legislators who served during a certain term (character)
district  legislative district (character)
party  democratic or republican (character)
fields  You can request specific fields by supplying a vector of fields names. Many fields are not returned unless requested. If you don’t supply a fields parameter, you will get the most commonly used subset of fields only. To save on bandwidth, parsing time, and confusion, it’s recommended to always specify which fields you will be using.
as  (character) One of table (default), list, or response (crl::HttpClient)
key  your SunlightLabs API key; loads from .Rprofile
...

Value

List of output fields.

Examples

## Not run:

```r
os_legislatorsearch(state = 'ca', party = 'democratic')
os_legislatorsearch(state = 'tx', party = 'democratic', active = TRUE)
os_legislatorsearch(state = 'nv', party = 'republican')
os_legislatorsearch(state = 'dc', chamber = 'upper')
```

# pass in more than one value for some parameters

```r
os_legislatorsearch(state = c('dc', 'or'), chamber = 'upper')
os_legislatorsearch(first_name = c('jane', 'bob'), chamber = 'upper')
```

## End(Not run)

---

**os_statemetasearch**  
**Search OpenStates metadata**

Description

Search OpenStates metadata
Usage

```r
os_statemetasearch(state = NULL, key = NULL, ...)
```

Arguments

- **state**: One or more two-letter state abbreviations (character)
- **key**: your SunlightLabs API key; loads from environment variable from .Renviron or from an option from .Rprofile
- **...**: Optional additional curl options (debugging tools mostly) passed on to `cru::HttpClient`

Value

A list with metadata for each state.

Examples

```r
## Not run:
os_statemetasearch()
os_statemetasearch('ca')
os_statemetasearch('tx')
os_statemetasearch('tx', verbose = TRUE)
```

---

## rsunlight-defunct

### Defunct functions in rsunlight

#### Description

- `ie_politicians`: Removed. See Influence Explorer below
- `ie_organizations`: Removed. See Influence Explorer below
- `ie_industries`: Removed. See Influence Explorer below
- `ie_individuals`: Removed. See Influence Explorer below
- `ie_lobbying`: Removed. See Influence Explorer below
- `ie_grants`: Removed. See Influence Explorer below
- `ie_contracts`: Removed. See Influence Explorer below
- `ie_faca`: Removed. See Influence Explorer below
- `ie_epa`: Removed. See Influence Explorer below
- `ie_entities`: Removed. See Influence Explorer below
- `ie_earmarks`: Removed. See Influence Explorer below
- `ie_contr_bundled`: Removed. See Influence Explorer below
- `ie_contr`: Removed. See Influence Explorer below

#### Influence Explorer

The Influence Explorer API is now defunct/gone.
sessions

Congressional sessions, up to the 114th (2015-2016)

Description

Congressional sessions, up to the 114th (2015-2016)

statements

statements methods

Description

statements methods

Usage

cg_statements_search(query, key = NULL, as = "table", ...)
cg_statements_recent(key = NULL, as = "table", ...)
cg_statements_subjects(key = NULL, as = "table", ...)
cg_statements_date(date, key = NULL, as = "table", ...)
cg_statements_subject(subject, key = NULL, as = "table", ...)
cg_statements_member(member_id, congress, key = NULL, as = "table", ...)

Arguments

query (character) a search term
key your ProPublica API key; pass in or loads from environment variable stored as PROPUBLICA_API_KEY in either your .Renviron, or similar file located in your home directory
as (character) IGNORED FOR NOW
... optional curl options passed on to curl::HttpClient. See curl::curl_options()
date (character) a date, of the form YYYY-MM-DD
subject (character) slug version of subject
member_id (character) The ID of the member to retrieve; it is assigned by the Biographical Directory of the United States Congress (http://bioguide.congress.gov/biosearch/biosearch.asp) or can be retrieved from a cg_members() request
congress (character) The number of the Congress this update took place during.
Value

various things for now, since return objects vary quite a bit among the different votes routes

Examples

```r
## Not run:
cg_statements_recent()
cg_statements_subjects()
cg_statements_search("AHCA")
cg_statements_date("2017-01-03")
cg_statements_subject('immigration')
cg_statements_member('C001084', 115)
```

### End(Not run)

<table>
<thead>
<tr>
<th>votes methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>votes methods</td>
</tr>
</tbody>
</table>

Description

votes methods

Usage

cg_vote(congress, chamber, session, id, key = NULL, as = "table", ...)

cg_votes_recent(chamber, key = NULL, as = "table", ...)

cg_votes_type(congress, chamber, type, key = NULL, as = "table", ...)

cg_votes_date(chamber, year = NULL, month = NULL, start_date = NULL, end_date = NULL, key = NULL, as = "table", ...)

cg_votes_senatenoms(congress, key = NULL, as = "table", ...)

cg_votes_explanations(congress, votes = FALSE, key = NULL, as = "table", ...)

cg_votes_explanations_category(congress, category, key = NULL, as = "table", ...)

cg_votes_explanations_member(member_id, congress, votes = FALSE, category = NULL, key = NULL, as = "table", ...)

cg_votes_explanations_member(member_id, congress, votes = FALSE, category = NULL, key = NULL, as = "table", ...)
Arguments

congress (character) The number of the Congress this update took place during.
chamber (character) The chamber this update took place in. 'house' or 'senate'.
session (integer) 1 or 2, depending on year (1 is odd-numbered years, 2 is even-numbered years)
id (integer) roll call number, i.e., vote id
key your ProPublica API key; pass in or loads from environment variable stored as PROPUBLICA_API_KEY in either your .Renviron, or similar file located in your home directory
as (character) IGNORED FOR NOW
... optional curl options passed on to curl::HttpClient. See curl::curl_options()
type (character) the vote type, one of missed, party, loneno or perfect
year (integer) a four digit year of the form YYYY
month (integer) a 1 or 2 digit month
start_date, end_date (character) start and end date, of the form YYYY-MM-DD
votes (logical) whether to return votes data or not. default: FALSE
category attribute describing the general reason for the absence or incorrect vote. see Details.
member_id (character) The ID of the member to retrieve; it is assigned by the Biographical Directory of the United States Congress (http://bioguide.congress.gov/biosearch/biosearch.asp) or can be retrieved from a cg_members() request

Value

various things for now, since return objects vary quite a bit among the different votes routes

Categories for the category parameter

- voted-incorrectly: Voted yes or no by mistake
- official-business: Away on official congressional business
- ambiguous: No reason given
- travel-difficulties: Travel delays and issues
- personal: Personal or family reason
- claims-voted: Vote made but not recorded
- medical: Medical issue for lawmaker (not family)
- weather: Inclement weather
- memorial: Attending memorial service
- misunderstanding: Not informed of vote
- leave-of-absence: Granted leave of absence
- prior-commitment: Attending to prior commitment
- election-related: Participating in an election
- military-service: Military service
- other: Other
Examples

## Not run:

```r
cg_vote(115, "senate", 1, 17)
cg_votes_recent("senate")
cg_votes_type(114, "house", "missed")
cg_votes_date("senate", year = 2017, month = 1)
cg_votes_date("senate", year = 2017, month = 3)
cg_votes_date("senate", start_date = "2017-01-03", end_date = "2017-01-31")
cg_votes_senatenoms(114)
cg_votes_explanations(114, votes = FALSE)
cg_votes_explanations(114, votes = TRUE)
cg_votes_explanations_category(114, "voted-incorrectly")
cg_votes_explanations_member("S001193", 115, votes = FALSE)
cg_votes_explanations_member("S001193", 115, votes = TRUE, category = "personal")
```

## End(Not run)
Index

*Topic datasets
  sessions, 26
*Topic package
  rsunlight-package, 2

cg_bill, 3
cg_bill_amendments, 7
cg_bill_cosponsors, 8
cg_bill_related_bills, 9
cg_bill_subjects, 9
cg_bill_subjects_search, 10
cg_bills, 4
cg_bills_recent, 4
cg_bills_recent_member, 5
cg_bills_recent_subject, 6
cg_bills_upcoming, 7
cg_committee (committees), 11
cg_committee_comms (committees), 11
cg_committee_hearings (committees), 11
cg_committees (committees), 11
cg_committees_comms_category (committees), 11
cg_committees_comms_chamber (committees), 11
cg_committees_comms_date (committees), 11
cg_floor_actions_date (floor_actions), 13
cg_floor_actions_recent (floor_actions), 13
cg_member (members), 14
cg_members (members), 14
cg_members (), 5, 26, 28
cg_members_bill_cosponsors (members), 14
cg_members_compare_bill_sponsors (members), 14
cg_members_compare_votes (members), 14
cg_members_leaving (members), 14
cg_members_new (members), 14
cg_members_state_district (members), 14
cg_members_votes (members), 14
cg_nomination (nominations), 15
cg_nominations_category (nominations), 15
cg_nominations_state (nominations), 15
cg_state_party_counties, 11
cg_statements_date (statements), 26
cg_statements_member (statements), 26
cg_statements_recent (statements), 26
cg_statements_search (statements), 26
cg_statements_subject (statements), 26
cg_statements_subjects (statements), 26
cg_vote (votes), 27
cg_votes_date (votes), 27
cg_votes_explanations (votes), 27
cg_votes_explanations_category (votes), 27
cg_votes_explanations_member (votes), 27
cg_votes_recent (votes), 27
cg_votes_senatenoms (votes), 27
cg_votes_type (votes), 27
committees, 11
curl::HttpClient, 12–14, 16, 18–26, 28
curl::curl_options (), 12–14, 26, 28

elements, 13

HttpClient, 3–11, 15

members, 14

nominations, 15

os_billlookup, 16
os_billsearch, 17
os_committee, 19
os_commitees, 19
os_district, 20
os districts, 21
os_legislator, 22
os_legislatorgeo, 23
INDEX

os_legislatorsearch, 23
os_statemetasearch, 24

rsunlight (rsunlight-package), 2
rsunlight-defunct, 25
rsunlight-package, 2

sessions, 26
statements, 26

votes, 27