
pyopenstates Documentation

Release 1.2.0

pyopenstates

Jun 04, 2018

Contents

1	Features	3
2	Installation	5
3	API Keys	7
3.1	pyopenstates module	7
3.2	Data structures	11
4	Indices and tables	17
	Python Module Index	19

A Python client for the [Open States API](#).

The [Open States](#) project provides data on legislators, bills, committees, and districts in legislatures of all 50 US states, plus D.C. and Puerto Rico. This data is gathered directly from the legislatures, and converted to a common format for interested developers, through a JSON API and data dumps. The project was originally created by the [Sunlight Foundation](#) as a part of Sunlight Labs. The Sunlight Foundation has since [shut down](#) Sunlight Labs, and Open States is now an [independent project](#).

This module is intended to be a replacement for the `openstates` methods provided by the original `sunlight` package. However, it is not a drop-in replacement; the methods are slightly different, and some features have been added.

Please consider [donating](#) or [volunteering](#) to support the Open States project. The data provided to the public by various legislatures can frequently change format. Without regularly-maintained scrapers and server infrastructure, this consistent and free API would not be possible.

CHAPTER 1

Features

- Compatible with Python 2.7-3+
- Full Unicode support
- Supports all methods, options, and data types provided by the API
- Methods for downloading data dumps
- Automatic conversion of string dates and timestamps to `datetime` objects
- Tested releases

CHAPTER 2

Installation

pyopenstates can be installed using [pip](#). Installation in a [virtualenv](#) is recommended.

To install the latest release, run:

```
$ pip install -U pyopenstates
```

Or, install the latest commit from git, run:

```
$ pip install -U git+https://github.com/openstates/pyopenstates
```


To use the Open States API you must obtain an [API Key](#).

Once you have your key you can use it with this library by setting the `OPENSTATES_API_KEY` environment variable or calling `pyopenstates.set_api_key()`.

Contents:

3.1 pyopenstates module

A Python client for the Open States API

exception `pyopenstates.APIError`
Raised when the Open States API returns an error

exception `pyopenstates.NotFound`
Raised when the API cannot find the requested object

`pyopenstates.download_bulk_data` (*state*, *file_object*, *data_format='json'*)
Downloads a zip containing bulk data on a given state to a given file object

Parameters

- **state** – The abbreviation of the state
- **file_object** – A file or file-like object
- **data_format** – `json` or `csv`

Note: `json` format provides much more detail than `csv`.

Examples

```
# Saving Ohio's data to a file on disk
with open("ohio-json.zip", "wb") as ohio_zip_file:
    pyopenstates.bulk_download("OH", ohio_zip_file)

# Or download it to memory
from io import BytesIO
mem_zip = BytesIO()
pyopenstates.bulk_download("OH", mem_zip)
```

`pyopenstates.get_bill` (*uid=None, state=None, term=None, bill_id=None, **kwargs*)

Returns details of a specific bill Can be identified by the Open States unique bill id (*uid*), or by specifying the state, term, and legislative bill ID

Parameters

- **uid** – The Open States unique bill ID
- **state** – The postal code of the state
- **term** – The legislative term (see state metadata)
- **bill_id** – The legislative bill ID (e.g. HR 42)
- ****kwargs** – Optional keyword argument options, such as `fields`,
- **specifies the fields to return** (*which*) –

Returns The *Bill* details as a dictionary

`pyopenstates.get_committee` (*com_id, fields=None*)

Gets committee details

Parameters

- **com_id** – Open States committee ID
- **fields** – An optional, custom set of fields to return

Returns The requested *Committee* details as a dictionary

`pyopenstates.get_district` (*boundary_id, fields=None*)

Gets district details

Parameters

- **boundary_id** – The boundary ID
- **fields** – Optionally specify a custom list of fields to return

Returns The requested *District* details as a dictionary

`pyopenstates.get_event` (*event_id, fields=None*)

Gets event details

Parameters

- **event_id** – The Open States Event UUID
- **fields** – An optional list of fields to return

Returns The requested *Event* details as a dictionary

`pyopenstates.get_legislator` (*leg_id, fields=None*)

Gets a legislator's details

Parameters

- **leg_id** – The Legislator’s Open States ID
- **fields** – An optional custom list of fields to return

Returns The requested *Legislator* details as a dictionary

`pyopenstates.get_metadata (state=None, fields=None)`

Returns a list of all states with data available, and basic metadata about their status. Can also get detailed metadata for a particular state.

Parameters

- **state** – The abbreviation of state to get detailed metadata on, or leave
- **None to get high-level metadata on all states. (as)** –
- **fields** – An optional list of fields to return; returns all fields by default

Returns The requested *Metadata* as a dictionary

Return type Dict

`pyopenstates.locate_legislators (lat, long, fields=None)`

Returns a list of legislators for the given latitude/longitude coordinates

Parameters

- **lat** – Latitude
- **long** – Longitude
- **fields** – An optional custom list of fields to return

Returns A list of matching *Legislator* dictionaries

`pyopenstates.search_bills (**kwargs)`

Find bills matching a given set of filters

Parameters ****kwargs** – One or more search filters

- **state** - Only return bills from a given state (e.g. nc)
- **chamber** - Only return bills matching the provided chamber

(upper or lower) - **bill_id** - Only return bills with a given bill_id. - **bill_id_in** - Accepts a pipe (|) delimited list of bill ids. - **q** - Only return bills matching the provided full text query. - **search_window** - By default all bills are searched, but if a time window is desired the following options can be passed to search_window:

- **search_window="all"** - Default, include all sessions.
- **search_window="term"** - Only bills from sessions within the

current term. - **search_window="session"** - Only bills from the current session. - **search_window="session:2009"** - Only bills from the session named 2009. - **search_window="term:2009-2011"** - Only bills from the sessions in the 2009-2011 session.

- **updated_since** - Only bills updated since a provided date (provided in

YYYY-MM-DD format) - `sponsor_id` Only bills sponsored by a given legislator id (e.g. ILL000555) - `subject` - Only bills categorized by Open States as belonging to this subject. - `type` Only bills of a given type (e.g. bill, resolution, etc.)

You can specify sorting using the following `sort` keyword argument values:

- `first`
- `last`
- `signed`
- `passed_lower`
- `passed_upper`
- `updated_at`
- `created_at`

Returns A list of matching *Bill* dictionaries

Note: This method returns just a subset (`state`, `chamber`, `session`, `subjects`, `type`, `id`, `bill_id`, `title`, `created_at`, `updated_at`) of the bill fields by default.

Use the `fields` parameter to specify a custom list of fields to return.

`pyopenstates.search_committees(**kwargs)`

Search for and return a list of matching committees

Parameters ****kwargs** – One or more filter keyword arguments

- `committee` - Name of committee.
- `subcommittee` - Name of subcommittee. (if None, object describes the

`committee`) - `chamber` - Chamber committee belongs to: upper, lower, or *‘joint’* - `state` - State abbreviation

Returns A list of matching *Committee* dictionaries

`pyopenstates.search_districts(state, chamber, fields=None)`

Search for districts

Parameters

- **state** – The state to search in
- **chamber** – the upper or lower legislative chamber
- **fields** – Optionally specify a custom list of fields to return

Returns A list of matching *District* dictionaries

`pyopenstates.search_events(**kwargs)`

Searches events

Events are not available in all states, to ensure that events are available check the `feature_flags` list in a states’ State Metadata.

Parameters ****kwargs** – One or more search filters

- `state` - State abbreviation.

- `type` - Categorized event type. (`committee:meeting` for now)

This method also allows specifying an alternate output format, by specifying `format=rss` or `format=ics`.

Returns A list of matching *Event* dictionaries

`pyopenstates.search_legislators(**kwargs)`

Search for legislators

Parameters ****kwargs** – One or more search filters

- `state` - Filter by state.
- `first_name` - Filter by first name.
- `last_name` - Filter by last name.
- `chamber` - Only legislators with a role in the specified chamber.
- `active` - True (default) to only include current legislators,

False will include all legislators - `term` - Only legislators that have a role in a certain term. - `district` - Only legislators that have represented the specified district. - `party` - Only legislators that have been associated with a specified party.

Returns A list of matching *Legislator* dictionaries

`pyopenstates.set_api_key(apikey)`

Sets API key. Can also be set as `OPENSTATES_API_KEY` environment variable.

`pyopenstates.set_user_agent(user_agent)`

Appends a custom string to the default User-Agent string (e.g. `pyopenstates/___version___user_agent`)

3.2 Data structures

Objects from the Open States API are returned as dictionaries

3.2.1 Metadata

- `abbreviation` - The two-letter abbreviation of the state.
- `capitol_timezone` - Timezone of state capitol (e.g. `America/New_York`)
- **chambers** - Dictionary mapping chamber type (upper/lower) to an object with the following fields:
 - `name` - Short name of the chamber (e.g. `House`, `Senate`)
 - `title` - Title of legislators in this chamber (e.g. `Senator`)
- **feature_flags** - A list of which optional features are available, options include:
 - `subjects` - bills have categorized subjects
 - `influenceexplorer` - legislators have influence explorer ids
 - `events` - event data is present
- `latest_csv_date` - Date that the CSV file at `latest_csv_url` was generated.
- `latest_csv_url` - URL from which a CSV dump of all data for this state can be obtained.

- `latest_json_date` - Date that the JSON file at `latest_json_url` was generated.
- `latest_json_url` - URL from which a JSON dump of all data for this state can be obtained.
- `latest_update` - Last time a successful scrape was run.
- `legislature_name` - Full name of legislature (e.g. North Carolina General Assembly)
- `legislature_url` - URL to legislature's official website.
- `name` - Name of state.
- **`session_details` - Dictionary of session names to detail dictionaries with the following keys:**
 - `type` - primary or special
 - `display_name` - e.g. 2009-2010 Session
 - `start_date` - date session began
 - `end_date` - date session began
- **`terms` - List of terms in order that they occurred. Each item in the list is comprised of the following keys:**
 - `start_year` - Year session started.
 - `end_year` - Year session ended.
 - `name` - Display name for term (e.g. 2009-2011).
 - `sessions` - List of sessions (e.g. 2009). Each session will be present in `session_details`.

3.2.2 Bill

- `state` - State abbreviation.
- `session` - Session key (see State Metadata for details).
- `bill_id` - The official id of the bill (e.g. SB 27, A 2111)
- `title` - The official title of the bill.
- `alternate_titles` - List of alternate titles that the bill has had (Often empty).
- **`action_dates` - Dictionary of notable action dates (useful for determining status). Contains the following fields:**
 - `first` - First action (only None if there are no actions).
 - `last` - Last action (only None if there are no actions).
 - `passed_lower` - Date that the bill seems to have passed the lower chamber (might be None).
 - `passed_upper` - Date that the bill seems to have passed the upper chamber (might be None).
 - `signed` - Date that the bill appears to have signed into law (might be None).
- `actions` - List of objects representing every recorded action for the bill. Action objects have the following fields:
 - `date` - Date of action.
 - `action` - Name of action as state provides it.
 - `actor` - The chamber, person, committee, etc. responsible for this action.
 - `type` - Open States-provided action categories, see action categorization.

- `chamber` - The chamber of origination ('upper' or 'lower')
- `created_at` - The date that this object first appeared in our system. (Note: not the date of introduction, see `action_dates` - for that information.)
- `updated_at` - The date that this object was last updated in our system. (Note: not the last action date, see `action_dates` for that information.)
- `documents` List - of associated documents, see versions for field details.
- `id` - Open States-assigned permanent ID for this bill.
- `scraped_subjects` - List of subject areas that the state categorized this bill under.
- `subjects`` - List of Open States standardized bill subjects, see subject categorization.
- `sources` - List of source URLs used to compile information on this object.
- **sponsors - List of bill sponsors.**
 - `name` - Name of sponsor as it appears on state website.
 - `leg_id` - Open States assigned legislator ID (will be `None` if no match was found).
 - `type` - Type of sponsor (`primary` or `cosponsor`)
- `type` - List of bill types.
- **versions Versions of the bill text. Both documents and versions have the following fields:**
 - `url` - Official URL for this document.
 - `name` - An official name for this document.
 - `mimetype` - The mimetype for the document (e.g. `text/html`)
 - `doc_id` - An Open States-assigned id uniquely identifying this document.
- **votes - List of vote objects. A vote object consists of the following keys:**
 - `motion` - Name of motion being voted upon (e.g. `Passage`)
 - `chamber` - Chamber vote took place in (`upper`, `lower`, `joint`)
 - `date` - Date of vote.
 - `id` - Open States-assigned unique identifier for vote.
 - `state` - State abbreviation.
 - `session` - Session key (see State Metadata for details).
 - `sources` - List of source URLs used to compile information on this object. (Can be empty if vote shares sources with bill.)
 - `yes_count` - Total number of yes votes.
 - `no_count` - Total number of no votes.
 - `other_count` - Total number of other votes (`abstain`, `not present`, etc.).
 - `yes_votes`, `no_votes`, `other_votes` - List of roll calls of each type. Each is an object consisting of two keys:
 - * `name` - Name of voter as it appears on state website.
 - * `leg_id` - Open States assigned legislator ID (will be `None` if no match was found).

3.2.3 Legislator

- `leg_id` - Legislator's permanent Open States ID. (e.g. ILL000555, NCL000123)
- `state` - Legislator's state.
- `active` - Boolean value indicating whether or not the legislator is currently in office.
- `chamber` - Chamber the legislator is currently serving in if active (`upper` or `lower`)
- `district` - District the legislator is currently serving in if active (e.g. 7, 6A)
- `party` - Party the legislator is currently representing if active.
- `email` - Legislator's primary email address.
- `full_name` - Full display name for legislator.
- `first_name` - First name of legislator.
- `middle_name` - Middle name of legislator.
- `last_name` - Last name of legislator.
- `suffixes` - Name suffixes (e.g. Jr., III) of legislator.
- `photo_url` - URL of an official photo of this legislator.
- `url` - URL of an official website for this legislator.
- `created_at` - The date that this object first appeared in our system.
- `updated_at` - The date that this object was last updated in our system.
- `created_at` - Date at which this legislator was added to our system.
- `updated_at` - Date at which this legislator was last updated.
- **offices** - List of office objects representing contact details for the legislator. Comprised of the following fields:
 - `type` - `capitol` or `district`
 - `name` - Name of the address (e.g. 'Council Office', 'District Office')
 - `address` - Street address.
 - `phone` - Phone number.
 - `fax` - Fax number.
 - `email` - Email address. Any of these fields may be `None` if not found.
- `roles` - List of currently active role objects if legislator is in office.
- `old_roles` - Dictionary mapping term keys to lists of roles that were valid for that term.

Role

- `term` - Term key for this role. (See metadata notes on terms and sessions for details.)
- `chamber`
- `state`
- `start_date` (optional)
- `end_date` (optional)

- `type` - member or committee member

If the role `type` is `member`:

- `party`
- `district`

And if the `type` is `committee member`:

- `committee` - Name of parent committee
- `subcommittee` - Name of subcommittee (if `None`, membership is just for a committee)
- `committee_id` - Open States id for committee that legislator is a member of
- `position` - Position on committee
- `old_roles` - List of old roles
- `sources` List of URLs used in gathering information for this legislator.

3.2.4 Committee

- `id` - Open States assigned committee ID.
- `state` - State abbreviation.
- `chamber` - Chamber committee belongs to: upper, lower, or joint.
- `committee` - Name of committee.
- `subcommittee` - Name of subcommittee. (if `None`, object describes the committee)
- `parent_id` - Committee id pointing to the parent committee if this is a subcommittee.
- `sources` - List of URLs used in gathering information for this legislator.
- `created_at` - The date that this object first appeared in our system.
- `updated_at` - The date that this object was last updated in our system.
- **members** - List of member objects, each has the following keys:
 - `name` - Name of legislator as provided by state source.
 - `leg_id` - Open States-assigned legislator id. (`None` if no match found).
 - `role` - Member's role on the committee (e.g. `chair`, `vice-chair`; default role is `member`)

3.2.5 Event

- `id` - Open States assigned event ID.
- `state` - State abbreviation.
- `type` - Categorized event type. (`committee:meeting` for now)
- `description` - Description of event from state source.
- `documents` - List of related documents.
- `Location` - Location if known, as given by state (it is often just a room number).
- `when` - Time event begins.
- `end` - End time (`None` if unknown).

- `timezone` - Timezone event occurs in (e.g. America/Chicago).
- **participants** - List of participant objects, consisting of the following fields:
 - `chamber` - Chamber of participant.
 - `type` - Type of participants (legislator, committee)
 - `participant` - String representation of participant (e.g. Housing Committee, Jill Smith)
 - `id` - Open States ID for participant if a match was found (e.g. TXC000150, MDL000101)
 - `type` - What role this participant played (will be host, chair, participant).
- **related_bills** - List of related bills for this event. Comprised of the following fields:
 - `type` - Type of relationship (e.g. consideration)
 - `description` - Description of how the bill is related given by the state.
 - `bill_id` - State's bill id (e.g. HB 273)
 - `id` - Open States assigned bill id (e.g. TXB00001234)
- `sources` List of URLs used in gathering information for this legislator.
- `created_at` - The date that this object first appeared in our system.
- `updated_at` - The date that this object was last updated in our system.

3.2.6 District

- `abbr` - State abbreviation.
- `boundary_id` - `boundary_id` used in District Boundary Lookup
- `chamber` - Whether this district belongs to the upper or lower chamber.
- `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.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

p

pyopenstates, 7

A

APIError, 7

D

download_bulk_data() (in module pyopenstates), 7

G

get_bill() (in module pyopenstates), 8

get_committee() (in module pyopenstates), 8

get_district() (in module pyopenstates), 8

get_event() (in module pyopenstates), 8

get_legislator() (in module pyopenstates), 8

get_metadata() (in module pyopenstates), 9

L

locate_legislators() (in module pyopenstates), 9

N

NotFound, 7

P

pyopenstates (module), 7

S

search_bills() (in module pyopenstates), 9

search_committees() (in module pyopenstates), 10

search_districts() (in module pyopenstates), 10

search_events() (in module pyopenstates), 10

search_legislators() (in module pyopenstates), 11

set_api_key() (in module pyopenstates), 11

set_user_agent() (in module pyopenstates), 11