

---

# **Zot\_Bib\_Web Documentation**

*Release 2.1.0*

**David Reitter**

**Feb 14, 2018**



---

## Contents:

---

<b>1</b>	<b>Example Installations</b>	<b>3</b>
<b>2</b>	<b>Documentation</b>	<b>5</b>
<b>3</b>	<b>Author/Contact:</b>	<b>7</b>
<b>4</b>	<b>Features</b>	<b>9</b>
<b>5</b>	<b>Demo</b>	<b>11</b>
<b>6</b>	<b>License and Donations</b>	<b>13</b>
<b>7</b>	<b>Requirements</b>	<b>15</b>
<b>8</b>	<b>Setup</b>	<b>17</b>
<b>9</b>	<b>Bibliography in Zotero</b>	<b>19</b>
<b>10</b>	<b>Overview of Configuration options</b>	<b>21</b>
<b>11</b>	<b>Deployment to a web site</b>	<b>23</b>
<b>12</b>	<b>Wordpress Support</b>	<b>25</b>
<b>13</b>	<b>Running the zot.py program</b>	<b>27</b>
	13.1 Positional Arguments . . . . .	27
	13.2 Named Arguments . . . . .	27
<b>14</b>	<b>Settings files</b>	<b>29</b>
<b>15</b>	<b>Indices and tables</b>	<b>33</b>
	<b>Python Module Index</b>	<b>35</b>



## Bibliography

Type [Journal Articles \(94\)](#) | [Archival Conference Publications \(11\)](#) | [Conference and Workshop Papers \(111\)](#) | [Books \(29\)](#) | [Book Chapters \(47\)](#) | [Edited Volumes \(8\)](#) | [Theses \(6\)](#) | [Tech Reports \(35\)](#) | [Talks \(11\)](#) | [Computer Programs \(2\)](#)

Topic [ACT-R \(16\)](#) | [Artificial Intelligence \(4\)](#) | [Computational Social Science \(5\)](#) | [Decision-Making \(2\)](#) | [Decision-Making and Cognitive Security \(7\)](#) | [General Cognitive Modeling \(21\)](#) | [General Cognitive Science \(14\)](#) | [Human-Computer Interaction \(22\)](#) | [Language \(32\)](#) | [Miscellaneous \(141\)](#) | [Other Topics \(1\)](#) | [Physiology \(21\)](#) | [Soar \(28\)](#) | [Social Cognition \(18\)](#) | [Software \(2\)](#) | [Talks \(Without Paper\) \(7\)](#) | [Technical Reports \(13\)](#)

Keywords

### Refereed Works

#### Language

Wang, Yale, David Reitter, and John Yen. "How Emotional Support and Informational Support Relate to Linguistic Alignment." *Social, Cultural, and Behavioral Modeling*. Washington, D.C.: Springer, 2017. Web. LNC3.  
[Abstract](#) [PDF](#) [BIB](#) [WordDoc](#) [EtcNotes](#)

Xu, Yang, and David Reitter. "Spectral Analysis of Information Density in Dialogue Predicts Collaborative Task Performance." *Proceedings of the 55th Annual Mtg. of the Association for Computational Linguistics*. Vancouver, Canada: N.p., 2017. Web.  
[Abstract](#) [PDF](#) [BIB](#) [WordDoc](#) [EtcNotes](#)

Kelly, Matthew A., David Reitter, and Robert L. West. "Degrees of Separation in Semantic and Syntactic Relationships." *Proc 15th International Conference on Cognitive Modeling*. Warwick, UK: N.p., 2017. Print.  
[Abstract](#) [BIB](#) [WordDoc](#) [EtcNotes](#)

Cole, Jeremy, Mojgan Ghafurian, and David Reitter. "Linking Memory Activation and Word Adoption in Social Language Use via Rational Analysis." *Proceedings of the 15th International Conference on Cognitive Modeling*. London, UK: N.p., 2017. Print.  
[Abstract](#) [BIB](#) [WordDoc](#) [EtcNotes](#)



# CHAPTER 1

---

## Example Installations

---

Lab website at Penn State

Personal website



## CHAPTER 2

---

### Documentation

---

<http://zot-bib-web.readthedocs.io/>



## CHAPTER 3

---

Author/Contact:

---

Dr. David Reitter, College of Information Sciences & Technology, Penn State [reitter@psu.edu](mailto:reitter@psu.edu)



## CHAPTER 4

---

### Features

---

This tool generates interactive web bibliographies based on one or more collections in a Zotero repository. Collections can be maintained by groups of people, using Zotero’s web interface or their desktop applications.

Bibliographies can be ordered by collection, by publication year, or by publication type (e.g., journal articles first), are interactively searchable, can be linked to PDF documents or other URLs, have records for BibTex, EndNote and Wikipedia, and can be exported to HTML or pushed to a Wordpress database.

Zot\_bib\_web does not depend on any third-party web server. The generated bibliographies load quickly because they are stored as static files along with the rest of your website. This makes a good source for webcrawlers, including Google Scholar and CiteSeer.

Setup is easy for anyone who runs their own website and knows how to use a command line (shell). The easiest way to use it is to call zot.py with the key of a public Zotero collection. It will make a zotero-bib.html file. Copy this, along with the “site” and “files” (if any) directory to your webserver.



## CHAPTER 5

---

### Demo

---

- View the HTML files in the demo folder for some examples of bibliographies. Their respective settings files and CSS style files are included.
- Run:

```
./zot.py --settings demo/settings3.py
```

to see it in action.



## CHAPTER 6

---

### License and Donations

---

- Use and modify this software free of charge.
- No warranty is provided whatsoever.
- Please e-mail [david.reitter@gmail.com](mailto:david.reitter@gmail.com) a link to the bibliography on your website if you decide to use zot\_bib\_web.
- You may use this software for free. A donation is suggested, either via Bitcoin (1CsboLGieSnxW-eVC4dFZjBGPQEn5Uyfsia) or with a credit card at <http://aquamacs.org/donate>



## CHAPTER 7

---

### Requirements

---

- Python 2.7 or 3.6+
- Pызotero. To install Pызotero, a library for python:

```
sudo pip install pызotero
```

or:

```
sudo easy_install pызotero
```

- A Zotero collection with your bibliography (as user or as group)
- Optional: dateutils package for Python (improves date parsing if present)



- Ensure zot.py is executable (`chmod ug+x zot.py`)
- Try it out. From a unix-like command-line, do this:

```
./zot.py --group 160464 DTDTV2EP
```

Then view `zotero-bib.html` in a browser. If that looks good, move on to the next steps for configuration.

- In a new file called `settings.py`, add configuration as documented in the file `settings_example.py`. Go to [zotero.org](http://zotero.org) to get your API secret key and your user or library IDs. It's easy: see the top of `settings.example.py` for details. If `settings.py` is set up, you can call `zot.py` without arguments.

Alternatively, you can use give the primary settings in arguments to the program.



---

## Bibliography in Zotero

---

- With Zotero, create a bibliography and note its ID (e.g., from the URL in the Zotero web interface). Example: MGID90AT. This ID is what you need for the “`toplevelfilter`” variable in `settings.py`.
- You can add sub-collections to your bibliography.
- If you format ordered by collections, giving them an order may be helpful. You can name collections starting with a number: “10 Social Psychology”.

Here’s an example of a bibliography structure:

```
My Publications [MGID90AT]
  10 Selected Works
  15 In Preparation / Under Review
  20 Refereed Works by Topic
    Semantics
    Parsing
    Dialogue
    Machine Learning
  30 Theses
  40 Talks (Without Paper)
```

To see this, use the provided `settings.py` as an example.



---

## Overview of Configuration options

---

- Configuration takes place in a settings file, by default named settings.py.
- Call `./zot.py -help` to see a list of command-line options.
- Please refer to the documentation for information on the settings file, or read settings\_example.py. A few options are discussed in the following.
- You can order our bibliography by sub-collection, by year, or by publication type (e.g., journal articles first, then conference papers). Even within the higher-level categories you can sort your bibliographic entries as you wish. Use the “sort\_criteria” and “show\_top\_section\_headings” settings.
- You can choose a different formatting convention. Default is APA format.
- At the top, a search box and a set of shortcuts may be displayed: for example, several years or spans of years, publication types, or subcollections that indicate topic areas. A visitor may click on these to quickly filter the bibliography.
- Configure the shortcuts shown at the top using the show\_shortcuts variable. ‘collection’, ‘type’, ‘year’, ‘venue’, ‘venue\_short’, and ‘tags’ are supported values, in addition to more fine-grained lists of values you can create using the shortcut() function. You can give the list of values or ranges (for years), their order, and some filtering to only show the most common ones. See settings.example.py for a detailed example.
- There are several more options. Again, see settings\_example.py.



---

## Deployment to a web site

---

- Upload the site folder or its contents to a public place on your web server. By default, /site/... is the assumed URL.

To generate HTML and include it in a website:

- run zot.py once/on demand, or install as cron job or service on a server Do not run it more than once a day. Configure it directly in zot.py, or in a separate file settings.py to make upgrading simple.
- include the resulting file zotero-bib.html (or as configured) in your website as you see fit. You may also include individual collection files, which are also generated. You can configure zot.py to generate a complete HTML document, or just a portion of it. Zot\_bib\_web generates HTML5 content.
- Style your bibliography using CSS. An example style file is included (see site/ directory).



This package can push directly to a Wordpress site. A separate program “push.py” is included to do this.

Follow these steps:

1. Set up zot.py to generate a bibliography you like. Call zot.py -full to generate a complete zotero-bib.html file for debugging purposes. Configure settings.py to not generate the full HTML code.
2. Install the wpautop-control plugin (or a similar plugin) to make sure that WP will not insert paragraph breaks at various places in the bibliography. With this plugin, you will need to add a “custom field” to the page created in the next step (Choose “Screen Options” at the top of the page view, enable custom fields. Then find custom fields at the very bottom of the page and add a “wpautop” field with value “no”).
3. Create a WP page or a post for the bibliography. Insert [zot\_bib\_web COLLECTION] where you’d like the bibliography inserted. Replace COLLECTION with the ID of the collection. (More options: see push.py)
4. Copy the style sheet contents (in site/) to your Wordpress theme (select “editor”, or “Additional CSS”).
5. Configure settings.py so that jquery and other files are available on the web server. Typically, this would be jquery\_path = “../wp-includes/js/jquery/jquery.js” clipboard.js and clippy.svg: You may refer to a public URL or serve the files yourself.
6. Configure push.py (at the top). You will need to know a few simple details about your WP installation.
7. Run push.py regularly or on demand. It will call zot.py automatically and then update the page in WP.



---

## Running the zot.py program

---

Add a fast, interactive Zotero bibliography to your website.

```
usage: Zot_Bib_Web [-h] [--settings SETTINGSFILE]
                  [--user USER | --group GROUP] [--api_key API_KEY]
                  [--output OUTPUT] [--verbose] [--quiet] [--div | --full]
                  [--no_cache]
                  [COLLECTION]
```

### 13.1 Positional Arguments

**COLLECTION**     Start at this collection

### 13.2 Named Arguments

<b>--settings, -s</b>	load settings from FILE. See settings_example.py.
<b>--user</b>	load a user library [user_library(...)]
<b>--group</b>	load a group library [group_library(...)]
<b>--api_key</b>	set Zotero API key [user_library(..., api_key=...)]
<b>--output, -o</b>	Output to this file [outputfile]
<b>--verbose</b>	output more information Default: 0
<b>--quiet</b>	output no information Default: 0
<b>--div</b>	output an HTML fragment [write_full_html_header=False]

**--full**                    output full html [write\_full\_html\_header=True]  
**--no\_cache, -n**           do not use cache [no\_cache]

---

## Settings files

---

The default name for a settings file is *settings.py*, but any settings file may be loaded using the *-settings* arguments.

See *settings\_example.py* for an example explanations.

**user\_collection** (*id*, *api\_key=None*, *collection=None*, *top\_level=False*)

Include collection from a user library in Zotero. See *group\_collection()*.

**group\_collection** (*id*, *api\_key=None*, *collection=None*, *top\_level=False*)

Include collection from a group library in Zotero.

Use *group\_collection()* for a group library, *user\_collection()* for a (private) user library. ID specifies the group or user ID.

You may find your user ID for the *library\_id* setting under “Settings -> Feeds/API”: <https://www.zotero.org/settings/keys>

You may find your library ID by selecting the group on the Zotero website, and then choose “Group Settings”. The URL in your browser window will then show you a six-digit number, e.g., *.../groups/110233/settings*

### Parameters

- **api\_key** (*str*) – The secret key provided by Zotero. If you want to retrieve non-public data from Zotero, you’ll need a Zotero account (or group) at [zotero.org](https://www.zotero.org). Log into your account, access the Settings page on the Zotero site and create an private API key (under “Settings -> Feeds/API”). For the key, check “Allow library access”. This key is used in the *api\_key* setting.
- **collection** (*str*) – ID of the top-level collection to be included. All sub-collections under this collection will be imported. If not given (None), all available collections will be included.
- **top\_level** (*str*) – If true, then the collection given be included as a level. Otherwise (default, False), sub-collections and items will be included directly.

It is recommended to make one collection in Zotero, for example, “website”, and then create titled sub-collections, like so:

```
toRead
thesis
website
  10 Selected Works
  20 Journal Articles
  30 Conference Proceedings
  40 Theses
```

The ID of the top-level collection called *website* is to be included as *collection* argument.

To find this ID: When you click on it on the Zotero website, your browser will show you an alphanumeric key in the URL, e.g., `items/collectionKey/FCQM2AY6`. The portion ‘FCQM2AY6’ is what you would use in ‘collection’ for the `user_collection()` or `group_collection()` directives.

Individual sub-collections may be excluded using `exclude_collection()`. Sub-collections may be renamed or merged using `rename_collection()`.

To cause `zot_bib_web` to format a sub-collection in special ways, you may add further statements, such as `featured_collection()`, `hidden_collection()`, `misc_collection()`, `short_collection()`.

**exclude\_collection** (*collection*, *top\_level\_only=False*)

Remove sub-collection *collection*. If *top\_level\_only* is True, only exclude this collection and items directly under it, but not its sub-collections.

**rename\_collection** (*collection*, *newName*)

Rename collection *collection* to *newName*. This may be used to merge collections by giving them the same name.

**short\_collection** (*collection*)

Short mode *collection*. This sub-collection will be shown using titles, journal and years only, which can then be expanded. Journal or conference titles can be kept short. Specify the “journal abbr or “conference title” fields, or a short “note” if necessary. You may want to copy bibliographic items from other parts of the bibliography into this sub-collection. You may also use a ‘\*’ before the name of the collection in the library.

**featured\_collection** (*collection*)

Feature *collection*. Extract this sub-collection and show at the beginning of the bibliography, regardless of whether the rest of the bibliography is sorted by, e.g., year, and ignores the collections otherwise. In the collection shown below, it prevents “in review” articles to show up as regular journal articles (which might give the impression you’re taking credit for not-yet-reviewed/published material!) You may also use a ‘!’ before the name of the collection in the library.

**hidden\_collection** (*collection*)

Hide sub-collection *collection*. We still add a shortcut at the top to unhide its contents if they are available elsewhere. You may also use a ‘-’ before the name of the collection in the library.

**misc\_collection** (*collection*)

Show only new items in *collection*. Show items in this collection, but exclude those items that are already included in another regular collection. A regular collection is one that is not hidden, not short, and not featured. This is useful to add a “Miscellaneous” category at the end for additional items without duplicating anything. You may also use a ‘&’ before the name of the collection in the library.

**exclude\_items** (*filter*)

After all items are loaded, filter them using a function. The function given in *filter* takes one argument, ITEM, and returns True for each item to exclude. ITEM is of type ZotItem.

**titlestring = ‘Bibliography’**

The title shown for the bibliography document

**bib\_style = 'apa'**

Style. 'apa', 'mla', or any other style known to Zotero

**sort\_criteria = ['collection', '-year', 'type']**

List of strings giving a hierarchy of subsections and ordering within them. Possible values include 'collection', 'year', 'type'. Prepend an item with '-', e.g., '-year' to sort in descending order.

**show\_top\_section\_headings = 1**

Number of first sort\_criteria to show as section headings E.g., if 1, the first element from *sort\_criteria* will be shown as section heading, and the rest without section headings (but ordered).

**number\_bib\_items = False**

If True, enumerate bibliographic items within a category as a list.

**show\_shortcuts = ['collection']**

List of shortcuts. Permissible values include the strings 'collection', 'year', 'type', 'venue', and 'venue\_short', or objects made with the function *shortcut()*.

**shortcut** (*crit, values=None, topN=None, sortDir='auto', sortBy=None*)

Make a shortcut to the *show\_shortcuts* list.

#### Parameters

- **crit** (*str*) – The criterion as a string, selected by the shortcut. Permissible values include 'collection', 'year', 'type', 'venue', and 'venue\_short'.
- **values** (*list*) – Optional list of values to be show for the criterion. Each element may be string, or an int (if appropriate, for years). For numbers, strings may specify a range, e.g., "2004-2009" (to select the range of years), or "-2004" or "2010-" to select years before or after the given year, respectively.
- **topN** (*int*) – If given, only show the TOPN values with the most bibliographic entries.
- **sortDir** (*str*) – Direction of sorting. If given, 'asc' or 'desc', or None (to turn off sorting).
- **sortBy** (*str*) – May be given as 'count', which indicates sorting by the number of bibliographic entries covered by each value, or 'name', to sort by name. The canonical order is default.

**show\_links = ['abstract', 'url', 'BIB', 'Wikipedia', 'EndNote']**

List of Links. Possible values: 'abstract', 'url', 'BIB', 'Wikipedia', 'EndNote', 'RIS', 'MLA', 'Cite.MLA', 'Cite.APA', 'Cite.<STYLE>'

**omit\_COinS = False**

If True, do not include COinS metadata

**smart\_selections = True**

If True, prevent user from selecting/copying text that shouldn't be copied.

**outputfile = 'zotero-bib.html'**

The resulting HTML document will be in this file.

**write\_full\_html\_header = True**

If True, a standalone HTML file is written (default).

**stylesheet\_url = 'site/style.css'**

URL to the style file on the web server.

**jquery\_path = 'site/jquery.min.js'**

URL to jQuery on the server

**show\_copy\_button = True**

If True, show a button that copies text to clipboard.

**clipboard\_js\_path = 'site/clipboard.min.js'**

URL to Clipboard.min.js on the server.

**copy\_button\_path = 'site/clippy.svg'**

URL to clippy.svg on the server.

**show\_search\_box = True**

Show a search box

**content\_filter = {'bib': <function fix\_bibtex\_reference>}**

Content filter for viewable or downloadable bibliographic content. Dict mapping strings to functions. Currently, only the function `fix_bibtex_reference` is supported, which changes bibtex reference symbols to the format nameYEARfirstword, e.g. smith2000towards.

**no\_cache = False**

If True, avoid use of cache

**language\_code = 'en'**

Language code used for `sortkeyname_order` and `link_translations`. Define labels for article types and their ordering. Dict, keys are language codes (indicating target language), values are dicts mapping fields to lists. Fields indicate bib item fields such as 'type' or 'date'. In the Zotero database, these may be in `libraryCatalog` or `itemType`. Lists are lists ordered by sort order. Each list element is a tuple of the form (value, label), where value indicates a value appropriate for the field, and the label is what is shown for that value in section headings and shortcuts.

Example:

```
'en' -> 'type' -> [('journalArticle', 'Journal Articles'), ...]
'en' -> 'date' -> [('in preparation', 'in prep.'), ...]
```

Example:

```
sortkeyname_order['en']['type'] = [
('journalArticle', 'Journal Articles'),
('archivalConferencePaper', 'Archival Conference Publications'),
('conferencePaper', 'Conference and Workshop Papers'),
('book', 'Books'),
('bookSection', 'Book Chapters'),
('edited-volume', "Edited Volumes"),
('thesis', 'Theses'),
('report', 'Tech Reports'),
('attachment', 'Document'),
('webpage', 'Web Site'),
('presentation', 'Talks'),
('computerProgram', 'Computer Programs')]
```

**link\_translations = {'de': {'abstract': 'Abstrakt', 'pdf': 'Volltext'}}**

Internationalization of link buttons (see also `show_links`). Dict, keys are language codes (indicating target language), values are dicts giving translation lexicons. Translation lexicons translate from English (keys) to the target language.

## CHAPTER 15

---

### Indices and tables

---

- `genindex`
- `search`



**Z**

zot, 29



## B

bib\_style (in module zot), 30

## C

clipboard\_js\_path (in module zot), 32

content\_filter (in module zot), 32

copy\_button\_path (in module zot), 32

## E

exclude\_collection() (in module zot), 30

exclude\_items() (in module zot), 30

## F

featured\_collection() (in module zot), 30

## G

group\_collection() (in module zot), 29

## H

hidden\_collection() (in module zot), 30

## J

jquery\_path (in module zot), 31

## L

language\_code (in module zot), 32

link\_translations (in module zot), 32

## M

misc\_collection() (in module zot), 30

## N

no\_cache (in module zot), 32

number\_bib\_items (in module zot), 31

## O

omit\_COinS (in module zot), 31

outputfile (in module zot), 31

## R

rename\_collection() (in module zot), 30

## S

short\_collection() (in module zot), 30

shortcut() (in module zot), 31

show\_copy\_button (in module zot), 31

show\_links (in module zot), 31

show\_search\_box (in module zot), 32

show\_shortcuts (in module zot), 31

show\_top\_section\_headings (in module zot), 31

smart\_selections (in module zot), 31

sort\_criteria (in module zot), 31

stylesheet\_url (in module zot), 31

## T

titlestring (in module zot), 30

## U

user\_collection() (in module zot), 29

## W

write\_full\_html\_header (in module zot), 31

## Z

zot (module), 27, 29