

Package: zotFunc (via r-universe)

October 16, 2026

Title Zotero Integration Functions

Version 0.1.0

Description Download complete Zotero group libraries via the Zotero Web API v3 in any of Zotero's export formats, and keep the local copies up to date by re-downloading only groups that changed on the server.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL <https://github.com/rkrug/zotFunc>, <https://rkrug.github.io/zotFunc/>

BugReports <https://github.com/rkrug/zotFunc/issues>

Config/roxygen2/version 8.0.0

Imports httr2 (>= 1.1.0)

Suggests jsonlite, knitr, rmarkdown, testthat (>= 3.0.0), vcr (>= 2.0.0), withr

Config/testthat/edition 3

Depends R (>= 4.1.0)

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev

Repository <https://rkrug.r-universe.dev>

Date/Publication 2026-10-16 10:42:08 UTC

RemoteUrl <https://github.com/rkrug/zotFunc>

RemoteRef main

RemoteSha 3e0d0401de9f6d76c97f4f1054e4c9a55ac6630d

Contents

get_group	2
get_group_last_modified	4
get_group_version	5
get_groupids_from_user	5
get_groups_from_user	6
group_outdated	7
id_from_name	8
Index	10

get_group	<i>Get Zotero Group Data</i>
-----------	------------------------------

Description

This function retrieves data from a Zotero group using the Zotero API.

Usage

```
get_group(  
  group_id = NULL,  
  path = tempfile(),  
  output_format = NULL,  
  api_key = NULL,  
  update = TRUE,  
  force = FALSE,  
  overwrite = FALSE,  
  max_active = 4  
)
```

Arguments

group_id	The ID of the Zotero group. If NULL, it is read from the file .id in path (written by a previous download), so an existing download can be updated by giving only path.
path	The path to save the retrieved data.
output_format	The format of the output files. Supported are: • bibtex: BibTeX format • biblatex: BibLaTeX format • bookmarks: Firefox bookmarks in HTML format • coins: COinS format • csjson: Citation Style Language (CSL) JSON format • csv: Comma-separated values format • mods: MODS format

- refer: Refer/BibIX format
- rdf_bibliontology: RDF/Bibliographic Ontology format
- rdf_dc: RDF/Dublin Core format
- rdf_zotero: RDF/Zotero format
- ris: RIS format
- tei: TEI format
- wikipedia: Wikipedia citation templates format
- NULL: Zotero's own JSON (most complete, but slowest)

See the [Zotero documentation](#) for details of the formats. If not given, it is read from `.format` in `path` (written by a previous download), otherwise NULL.

<code>api_key</code>	API key for Zotero. Only needed for private groups.
<code>update</code>	If TRUE, an existing previous download in <code>path</code> is updated (see Details). If FALSE, an existing download is an error. Default: TRUE.
<code>force</code>	If TRUE, download even if the downloaded version equals the server version. If FALSE, only download if the server version is larger. Default: FALSE.
<code>overwrite</code>	If TRUE, an existing non-empty path that is not a previous download is deleted and replaced. If FALSE, this is an error. Default: FALSE.
<code>max_active</code>	Number of pages (of 100 records) downloaded in parallel, from 1 to 4 (the maximum number of concurrent requests Zotero asks for). Lower it if Zotero keeps answering with 429 Too Many Requests. Default: 4.

Details

This function uses the Zotero API to retrieve data from a Zotero group. It downloads the data in the format specified in `output_format` and saves it in batches of 100 records to the folder specified by the `path` parameter. Notes and attachments count as records for the API but are not exported in most formats, so pages without any exported record (empty, or only the header for csv) are not saved.

NB: An existing path that is not a previous download (no `.version`) is only deleted and replaced if `overwrite` = TRUE; otherwise an error is raised. An empty folder is used as is.

After the download, the group id and the library version at the start of the download are written to the files `.id` and `.version` (and the group name in `.name`) in `path`. If `path` already contains such a download, `group_outdated()` is used to check it: if the group differs from `group_id`, or the server version is older than the downloaded one, an error is raised; if the versions are identical nothing is downloaded unless `force` = TRUE.

The output format is stored in `.format` (json for `output_format` = NULL). If `output_format` is not given, it is read from there. If it is given and differs from `.format`, the group is downloaded again in the new format, even if it is up to date.

The first page is downloaded alone, as it gives the number of records. The remaining pages are downloaded in parallel, `max_active` at a time, following the [rate limits of the Zotero API](#): no more than 4 concurrent requests, a Backoff header pauses the download for the given number of seconds after the running requests, and a 429 or 503 response is retried after its Retry-After (or with exponential backoff), pausing all other requests as well.

For further information about the allowed formats and other details on the API see https://www.zotero.org/support/dev/web_api/v3/start.

Value

The path where the data is saved (also when nothing was downloaded).

Examples

```
# Download the Zotero group with ID 2352922 (IPBES IAS Assessment Bibliography) as JSON
# into the folder "zotero_data"

## Not run:
get_group(2352922, "zotero_data")

# later: update it using only the folder
get_group(path = "zotero_data")

## End(Not run)
```

```
get_group_last_modified
```

Get the Last Modified Date of a Zotero Group

Description

Queries the most recently modified item of a Zotero group and returns its modification time. Only items are considered (not collections or tags).

Usage

```
get_group_last_modified(group_id = 2352922, api_key = NULL)
```

Arguments

group_id	The ID of the Zotero group.
api_key	API key for Zotero. Only needed for private groups.

Value

The last modification time as a POSIXct (UTC), or NA if the group has no items.

Examples

```
## Not run:
get_group_last_modified(2352922)

## End(Not run)
```

get_group_version	<i>Get the Library Version of a Zotero Group</i>
-------------------	--

Description

Returns the Last-Modified-Version of a Zotero group: an integer that increases whenever anything in the group changes. Store it after a download and compare it with the current value to decide whether a re-download is necessary.

Usage

```
get_group_version(group_id = 2352922, api_key = NULL)
```

Arguments

group_id	The ID of the Zotero group.
api_key	API key for Zotero. Only needed for private groups.

Value

The library version as an integer.

Examples

```
## Not run:  
get_group_version(2352922)  
  
## End(Not run)
```

get_groupids_from_user	<i>Get all the ids from all Groups from a User</i>
------------------------	--

Description

Lists the groups of a Zotero user.

Usage

```
get_groupids_from_user(  
  zotero_user_name = "ipbes",  
  zotero_user_id = NULL,  
  api_key = NULL,  
  verbose = FALSE  
)
```

Arguments

zotero_user_name	Name of the user to download the groups from. If zotero_user_id is specified, not needed. Default: "ipbes".
zotero_user_id	Zotero user id to download the groups from. Default: obtained through <code>id_from_name()</code> . In scripts, give the id explicitly: <code>id_from_name()</code> parses the profile page, which may change.
api_key	Zotero API key, only needed for private groups.
verbose	logical. If TRUE, output is verbose

Details

NB: Only the first 25 groups are returned (the Zotero default page size).

Public groups work without an API key. For private groups, create a key with read access at [Settings > Security \(Create New Private Key\)](#), with *Allow library access*, *Allow group access* and *Default Group Permissions: Read Only*.

Value

Character vector of group ids, named with the group names.

get_groups_from_user *Get all Groups from a User*

Description

Downloads every group of a Zotero user with `get_group()`, each into the sub-folder <group name>_<group id> of path. The update rules of `get_group()` apply to each group. A group that fails prints its error and the next group continues.

Usage

```
get_groups_from_user(
  zotero_user_name = "ipbes",
  zotero_user_id = NULL,
  path = tempfile(),
  output_format = "rdf_zotero",
  api_key = Sys.getenv("ZOTERO_API_IPBES"),
  update = TRUE,
  force = FALSE,
  overwrite = FALSE,
  max_active = 4
)
```

Arguments

zotero_user_name	Name of the user to download the groups from. Default: "ipbes". If not given and path contains a .user_id file (written by a previous run), the user id is read from there, so an existing download can be updated by giving only path.
zotero_user_id	Zotero user id to download the groups from. Default: read from path/.user_id if present, otherwise obtained through the function <code>id_from_name</code> . Must match path/.user_id if both exist. In scripts, give the id explicitly: <code>id_from_name()</code> parses the profile page, which may change.
path	Folder to save the groups in.
output_format	The output_format of the files. See <code>get_group</code> . If not given, existing group downloads keep the format stored in their .format file; new groups use the default "rdf_zotero".
api_key	Zotero API key, only needed for private groups. Default: the environment variable ZOTERO_API_IPBES.
update	If FALSE, an existing path is an error. Otherwise passed to <code>get_group</code> : existing group downloads are updated. Default: TRUE.
force	Passed to <code>get_group</code> : download even if the group is not outdated. Default: FALSE.
overwrite	Passed to <code>get_group</code> : replace existing group folders that are not previous downloads. Default: FALSE.
max_active	Passed to <code>get_group</code> : number of pages downloaded in parallel (1 to 4). Groups are downloaded one after the other. Default: 4.

Details

The user id and name are stored in .user_id and .user_name in path, so later updates only need path.

Public groups work without an API key. For private groups, create a key with read access at [Settings > Security](#) (**Create New Private Key**, with *Allow library access*, *Allow group access* and *Default Group Permissions: Read Only*).

Value

NULL, invisibly. Called for the downloads.

group_outdated	<i>Check if a Downloaded Zotero Group is Outdated</i>
----------------	---

Description

Compares the library version stored in group_dir/.version (written by `get_group()`) with the current version on the Zotero server.

Usage

```
group_outdated(group_dir, group_id = NULL, api_key = NULL)
```

Arguments

group_dir Folder of a previous `get_group()` download (containing `.id`, `.name` and `.version`).

group_id Optional. If given, it must match the group stored in `group_dir/.id`.

api_key API key for Zotero. Only needed for private groups.

Value

A named logical, named with the group id: TRUE if the server version is larger than the downloaded one, FALSE if they are identical. An error is raised if the server version is *smaller* than the downloaded one, if `.id` / `.version` are missing, or if `group_id` does not match `.id`.

Examples

```
## Not run:
group_outdated("zotero_data")

## End(Not run)
```

id_from_name

Get Zotero User ID from Username

Description

This function retrieves a Zotero user ID from a given username.

Usage

```
id_from_name(username)
```

Arguments

username The Zotero username. **Zotero Usernames are case sensitive!**

Details

NB:The function relies on analysing the profile page. As this page might change, it is not recommended to use this function programmatically, but rather to retrieve the id and to use it hardcoded.

Value

The Zotero user ID as a string.

Examples

```
## Not run:  
id_from_name("ipbes")  
# "5760254"  
  
## End(Not run)
```

Index

`get_group`, [2](#)
`get_group()`, [6–8](#)
`get_group_last_modified`, [4](#)
`get_group_version`, [5](#)
`get_groupids_from_user`, [5](#)
`get_groups_from_user`, [6](#)
`group_outdated`, [7](#)
`group_outdated()`, [3](#)

`id_from_name`, [8](#)
`id_from_name()`, [6, 7](#)