

---

# reader Documentation

*Release 0.4*

lemon24

Jan 02, 2019



---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>API reference</b>	<b>7</b>
3.1	API . . . . .	7
<b>4</b>	<b>Command-line interface</b>	<b>13</b>
4.1	Command-line interface . . . . .	13
<b>5</b>	<b>Web application</b>	<b>17</b>
5.1	Web application . . . . .	17
<b>6</b>	<b>Deployment options</b>	<b>21</b>
6.1	Deployment . . . . .	21
<b>7</b>	<b>Plugins</b>	<b>23</b>
7.1	Plugins . . . . .	23
<b>8</b>	<b>Development</b>	<b>25</b>
8.1	Development . . . . .	25
<b>9</b>	<b>Changelog</b>	<b>27</b>
9.1	reader changelog . . . . .	27
<b>10</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>



A minimal feed reader.



# CHAPTER 1

---

## Features

---

- Stable and clearly documented API.
- Excellent test coverage.
- Minimal web interface.





## CHAPTER 2

---

### Usage

---

```
$ pip install reader
```

```
>>> from reader import Reader
>>>
>>> reader = Reader('db.sqlite')
>>> reader.add_feed('http://www.hellointernet.fm/podcast?format=rss')
>>> reader.update_feeds()
>>>
>>> entries = list(reader.get_entries())
>>> [e.title for e in entries]
['H.I. #108: Project Cyclops', 'H.I. #107: One Year of Weird', ...]
>>>
>>> reader.mark_as_read(entries[0])
>>>
>>> [e.title for e in reader.get_entries(which='unread')]
['H.I. #107: One Year of Weird', 'H.I. #106: Water on Mars', ...]
>>> [e.title for e in reader.get_entries(which='read')]
['H.I. #108: Project Cyclops']
```



If you are looking for information on a specific function, class, or method, this part of the documentation is for you.

### 3.1 API

This part of the documentation covers all the interfaces of reader.

#### 3.1.1 Reader Object

All of reader's functionality can be accessed through a *Reader* instance.

**class** `reader.Reader` (*path=None*)

A feed reader.

**Parameters** `path` (*str*) – Path to the reader database.

**Raises** *StorageError*

**add\_feed** (*feed*)

Add a new feed.

**Parameters** `feed` (*str* or *Feed*) – The feed URL.

**Raises**

- *FeedExistsError*
- *StorageError*

**remove\_feed** (*feed*)

Remove a feed.

Also removes all of the feed's entries.

**Parameters** `feed` (*str* or *Feed*) – The feed URL.

**Raises**

- *FeedNotFoundError*
- *StorageError*

**get\_feeds** (*sort*='title')

Get all the feeds.

**Parameters** **sort** (*str*) – How to order feeds; one of 'title' (by *user\_title* or *title*, case insensitive; default), or 'added' (last added first).

**Yields** *Feed* – Sorted according to sort.

**Raises** *StorageError*

**get\_feed** (*feed*)

Get a feed.

**Parameters** **feed** (*str* or *Feed*) – The feed URL.

**Returns** The feed if it exists, None if it doesn't.

**Return type** *Feed* or None

**Raises** *StorageError*

**set\_feed\_user\_title** (*feed*, *title*)

Set a user-defined title for a feed.

**Parameters**

- **feed** (*str* or *Feed*) – The feed URL.
- **title** (*str* or None) – The title, or None to remove the current title.

**Raises**

- *FeedNotFoundError*
- *StorageError*

**update\_feeds** (*new\_only*=False)

Update all the feeds.

**Parameters** **new\_only** (*bool*) – Only update feeds that have never been updated.

**Raises** *StorageError*

**update\_feed** (*feed*)

Update a single feed.

**Parameters** **feed** (*str* or *Feed*) – The feed URL.

**Raises**

- *FeedNotFoundError*
- *StorageError*

**get\_entries** (*which*='all', *feed*=None, *has\_enclosures*=None)

Get all or some of the entries.

**Parameters**

- **which** (*str*) – One of 'all', 'read', or 'unread'.
- **feed** (*str* or *Feed* or None) – Only return the entries for this feed.
- **has\_enclosures** (*bool* or None) – Only return entries that (don't) have enclosures.

**Yields** *Entry* – Last published/updated entries first.

**Raises**

- *FeedNotFoundError* – Only if *feed* is not None.
- *StorageError*

**mark\_as\_read**(*entry*)

Mark an entry as read.

**Parameters** *entry* (*tuple*(*str*, *str*) or *Entry*) – (feed URL, entry id) tuple.

**Raises**

- *EntryNotFoundError*
- *StorageError*

**mark\_as\_unread**(*entry*)

Mark an entry as unread.

**Parameters** *entry* (*tuple*(*str*, *str*) or *Entry*) – (feed URL, entry id) tuple.

**Raises**

- *EntryNotFoundError*
- *StorageError*

### 3.1.2 Data Objects

**class** reader.**Feed**(*url*, *updated=None*, *title=None*, *link=None*, *author=None*, *user\_title=None*)

Data type representing a feed.

**url**

The URL of the feed.

**updated**

The date the feed was last updated.

**title**

The title of the feed.

**link**

The URL of a page associated with the feed.

**author**

The author of the feed.

**user\_title**

User-defined feed title.

**class** reader.**Entry**(*id*, *updated*, *title=None*, *link=None*, *author=None*, *published=None*, *summary=None*, *content=()*, *enclosures=()*, *read=False*, *feed=None*)

Data type representing an entry.

**id**

Entry identifier.

**updated**

The date the entry was last updated.

**title**

The title of the entry.

**link**

The URL of a page associated with the entry.

**author**

The author of the feed.

**published**

The date the entry was first published.

**summary**

A summary of the entry.

**content**

Full content of the entry. A sequence of *Content* objects.

**enclosures**

External files associated with the entry. A sequence of *Enclosure* objects.

**read**

Whether the entry was read or not.

**feed**

The entry's feed.

**class** `reader.Content` (*value*, *type=None*, *language=None*)

Data type representing a piece of content.

**value**

The content value.

**type**

The content type.

**language**

The content language.

**class** `reader.Enclosure` (*href*, *type=None*, *length=None*)

Data type representing an external file.

**href**

The file URL.

**type**

The file content type.

**length**

The file length.

### 3.1.3 Exceptions

All exceptions that *Reader* explicitly raises inherit from *ReaderError*.

**exception** `reader.ReaderError`

Base for all public exceptions.

**exception** `reader.FeedError` (*url*)

A feed error occurred.

Subclass of *ReaderError*.

**url = None**

The feed URL.

**exception** `reader.FeedExistsError(url)`

Feed already exists.

Subclass of *FeedError*.

**exception** `reader.FeedNotFoundError(url)`

Feed not found.

Subclass of *FeedError*.

**exception** `reader.ParseError(url)`

An error occurred while getting/parsing feed.

The original exception should be chained to this one (e.\_\_cause\_\_).

Subclass of *FeedError*.

**exception** `reader.EntryError(url, id)`

An entry error occurred.

Subclass of *FeedError*.

**url = None**

The feed URL.

**id = None**

The entry id.

**exception** `reader.EntryNotFoundError(url, id)`

Entry not found.

Subclass of *EntryError*.

**exception** `reader.StorageError`

An exception was raised by the underlying storage.

The original exception should be chained to this one (e.\_\_cause\_\_).

Subclass of *ReaderError*.





---

## Command-line interface

---

### 4.1 Command-line interface

This part of the documentation covers the reader command-line interface.

---

**Note:** The CLI is not stable yet and might change without any notice.

---

---

**Note:** The command-line interface is optional, use the `cli` extra to install its dependencies:

```
pip install reader[cli]
```

---

Most commands need a database to work. The following are equivalent:

```
python -m reader --db /path/to/db some-command  
READER_DB=/path/to/db python -m reader some-command
```

If no database path is given, `~/ .config/reader/db.sqlite` is used (at least on Linux).

Add a feed:

```
python -m reader add http://www.example.com/atom.xml
```

Update all feeds:

```
python -m reader update
```

Start a local reader server at `http://localhost:8080/`:

```
python -m reader serve
```

## 4.1.1 Reference

### reader

```
reader [OPTIONS] COMMAND [ARGS]...
```

#### Options

**--db** <db>  
Path to the reader database. Defaults to /home/docs/.config/reader/db.sqlite.

**--plugin** <plugin>  
Import path to a plug-in. Can be passed multiple times.

#### Environment variables

##### READER\_DB

Provide a default for *--db*

##### READER\_PLUGIN

Provide a default for *--plugin*

### add

Add a new feed.

```
reader add [OPTIONS] URL
```

#### Options

**--update, --no-update**  
Update the feed after adding it.

**-v, --verbose**

#### Arguments

##### URL

Required argument

### list

List feeds or entries.

```
reader list [OPTIONS] COMMAND [ARGS]...
```

## entries

List all the entries.

Outputs one line per entry in the following format:

<feed URL> <entry link or id>

```
reader list entries [OPTIONS]
```

## feeds

List all the feeds.

```
reader list feeds [OPTIONS]
```

## remove

Remove an existing feed.

```
reader remove [OPTIONS] URL
```

## Options

**-v, --verbose**

## Arguments

**URL**

Required argument

## serve

Start a local HTTP reader server.

```
reader serve [OPTIONS]
```

## Options

**-h, --host <host>**  
The interface to bind to.

**-p, --port <port>**  
The port to bind to.

**-v, --verbose**

## update

Update one or all feeds.

If URL is not given, update all the feeds.

```
reader update [OPTIONS] [URL]
```

## Options

**--new-only, --no-new-only**

Only update new (never updated before) feeds.

**-v, --verbose**

## Arguments

**URL**

Optional argument

## 5.1 Web application

### 5.1.1 Screenshots

Main page

Feed page

Feeds page

[entries](#) [feeds](#)

add feed

## Entries

unread [read all](#)

has enclosures: [yes](#) [no](#) don't care

### [H.I. #112: Consistency Hobgoblins](#)

[Hello Internet](#)

23 days ago

mark as read

A new Hello Internet artifact is born, American state identities revisited, Banksy's shredding stunt, government and social media, daylight savings time disaster, Scrabble, and First Man. Sponsors: Squarespace: head to [squarespace.com/hello](https://squarespace.com/hello) for a free...

- [112.mp3](#) [with tags](#) audio/mpeg

Fig. 1: main page

[entries](#) [feeds](#)

add feed

## Entries for xkcd.com

[unread](#) read [all](#)

has enclosures: [yes](#) [no](#) don't care

mark all as unread

delete feed

update feed title

## [Airplanes and Spaceships](#)

[xkcd.com](#)

5 days ago

mark as unread

## [Kilogram](#)

[xkcd.com](#)

8 days ago

Fig. 2: feed page

[entries](#) [feeds](#)

add feed

## Feeds

### [Hello Internet](#)

[site](#)

delete feed

update feed title

### [xkcd.com](#)

[site](#)

delete feed

update feed title

Fig. 3: feeds page



---

## Deployment options

---

### 6.1 Deployment

For reader to actually be useful as a feed reader, feeds need to get updated and the web application needs to be served.

#### 6.1.1 Updating feeds

You can run the `update` command regularly to update feeds (e.g. every hour). Note that reader uses the ETag and Last-Modified headers, so, if supported by the the server, feeds will only be downloaded if they changed.

To avoid waiting too much for a new feed to be updated, you can run `update --new-only` more often (e.g. every minute); this will update only newly-added feeds.

You can achieve this using cron:

```
42 * * * * reader update -v 2>&1 >>"/tmp/$LOGNAME.reader.update.hourly.log"
* * * * * reader update -v --new-only 2>&1 >>"/tmp/$LOGNAME.reader.update.new.log"
```

If you are running reader on a personal computer, it might also be convenient to run `update` once immediately after boot:

```
@reboot      sleep 60; reader update -v 2>&1 >>"/tmp/$LOGNAME.reader.update.boot.log"
```

#### 6.1.2 Serving the web application

reader exposes a standard WSGI application as `reader.app.wsgi:app`. See the [Flask documentation](#) for more details on how to deploy it. The path to the reader database can be configured through the `READER_DB` environment variable.

An example uWSGI configuration file (probably not idiomatic, from [here](#)):

```
[uwsgi]
socket = /apps/reader/uwsgi/sock
manage-script-name = true
mount = /reader=reader.app.wsgi:app
plugin = python3
virtualenv = /apps/reader/
env = READER_DB=/data/www-data/reader.sqlite
```

You can also run the web application with the `serve` command. `serve` uses Werkzeug's development server, so it probably won't scale well past a single user.

If running on a personal computer, you can use cron to run `serve` at boot:

```
@reboot      sleep 60; reader serve -p 8080 2>&1 ) >>"/tmp/$LOGNAME.reader.serve.boot.
↪log"
```

Note that the web application has no authentication whatsoever; it is expected a server / middleware will provide that.

### 7.1 Plugins

**Note:** The plugin system/hooks are not stable yet and may change without any notice.

The plugins listed here are also not stable.

**Note:** Plugin loading machinery is optional, use the `plugins` extra to install its dependencies:

```
pip install reader[plugins]
```

There is experimental support of plugins.

To load plugins, set the `READER_PLUGIN` environment variable to the plugin entry point (e.g. `package.module:entry_point`); multiple entry points should be separated by one space:

```
READER_PLUGIN='first.plugin:entry_point second_plugin:main' \  
python -m reader some-command
```

Currently plugins are loaded through the CLI only (they won't be loaded when importing *Reader* or when using the web application).

#### 7.1.1 Existing plugins

##### `regex_mark_as_read`

Mark added entries of specific feeds as read if their title matches a regex.

To load:

```
READER_PLUGIN='reader.plugins.regex_mark_as_read:regex_mark_as_read' \
READER_PLUGIN_REGEX_MARK_AS_READ_CONFIG='examples/regex_mark_as_read_config.json' \
python -m reader update -v
```

READER\_PLUGIN\_REGEX\_MARK\_AS\_READ\_CONFIG should be a JSON file with this structure:

```
{
  "feed-url": ["first-regex", "second-regex"]
}
```

Implemented for <https://github.com/lemon24/reader/issues/79>.

### feed\_entry\_dedupe

Deduplicate entries for the same feed.

Duplicates are entries with the same title *and* summary/content.

If the old entry is read, the new one will be too. If the old entry is unread, it will be marked as read in favor of the new one.

To load:

```
READER_PLUGIN='reader.plugins.feed_entry_dedupe:feed_entry_dedupe' \
python -m reader update -v
```

Implemented for <https://github.com/lemon24/reader/issues/79>.

### tumblr\_gdpr

Accept Tumblr GDPR stuff.

Since May 2018, Tumblr redirects all new sessions to an “accept the terms of service” page, including RSS feeds (supposed to be machine-readable), breaking them.

This plugin “accepts the terms of service” on your behalf.

To load:

```
READER_PLUGIN='reader.plugins.tumblr_gdpr:tumblr_gdpr' \
python -m reader update -v
```

Implemented for <https://github.com/lemon24/reader/issues/67>.

## 8.1 Development

### 8.1.1 Goals

Goals:

- clearly documented API
- minimal web interface
- minimal CLI

Development should follow a [problem-solution](#) approach.

### 8.1.2 Testing

Run tests:

```
tox
```

or:

```
make
```

or:

```
python3 -m pytest --runslow
```

Start a local development server for the web-app:

```
READER_DB=db.sqlite FLASK_DEBUG=1 FLASK_APP=autoapp.py \  
flask run -h localhost -p 8080
```

### 8.1.3 Making a release

Making a release (from  $x$  to  $y == x + 1$ ):

- make sure all tests pass etc.
- bump versions in `docs/conf.py` and `src/reader/__init__.py` to  $y$
- update changelog with release version and date
- clean up dist/: `rm -rf dist/`
- build tarball and wheel: `python setup.py build sdist` and `python setup.py build bdist_wheel`
- push to GitHub
- wait for Travis / Codecov / Read the Docs builds to pass
- upload to test PyPI and check: `twine upload --repository-url https://test.pypi.org/legacy/ dist/*`
- upload to PyPI: `twine upload dist/*`
- tag release in GitHub
- build docs from latest and enable  $y$  docs version (should happen automatically after the first time)
- bump versions from  $y$  to  $(y + 1).dev0$ , add  $(y + 1)$  changelog section

### 9.1 reader changelog

#### 9.1.1 Version 0.4

Released 2019-01-02

- Support Python 3.7.
- Entry `content` and `enclosures` now default to an empty tuple instead of `None`. (#99)
- `get_feeds()` now sorts feeds by `user_title` or `title` instead of just `title`. (#102)
- `get_feeds()` now sorts feeds in a case insensitive way. (#103)
- Add `sort` argument to `get_feeds()`; allows sorting feeds by title or by when they were added. (#98)
- Allow changing the feed sort order in the web application. (#98)

#### 9.1.2 Version 0.3

Released on 2018-12-22

- `get_entries()` now prefers sorting by `published` (if present) to sorting by `updated`. (#97)
- Add `regex_mark_as_read` plugin (mark new entries as read based on a regex). (#79)
- Add `feed_entry_dedupe` plugin (deduplicate new entries for a feed). (#79)
- Plugin loading machinery dependencies are now installed via the `plugins` extra.
- Add a `plugins` section to the documentation.

### 9.1.3 Version 0.2

Released on 2018-11-25

- Factor out storage-related functionality into a separate interface. (#94)
- Fix `update --new-only` updating the same feed repeatedly on databases that predate `--new-only`. (#95)
- Add web application screenshots to the documentation.

### 9.1.4 Version 0.1.1

Released on 2018-10-21

- Fix broken `reader serve` command (broken in 0.1).
- Raise `StorageError` for unsupported SQLite configurations at `Reader` instantiation instead of failing at run-time with a generic `StorageError("sqlite3 error")`. (#92)
- Fix wrong submit button being used when pressing enter in non-button fields. (#69)
- Raise `StorageError` for failed migrations instead of an undocumented exception. (#92)
- Use `requests-mock` in parser tests instead of a web server (test suite run time down by ~35%). (#90)

### 9.1.5 Version 0.1

Released on 2018-09-15

- Initial release; public API stable.
- Support broken Tumblr feeds via the `tumblr_gdpr` plugin. (#67)



## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `search`



### r

`reader`, [23](#)  
`reader.plugins.feed_entry_dedupe`, [24](#)  
`reader.plugins.regex_mark_as_read`, [23](#)  
`reader.plugins.tumblr_gdpr`, [24](#)



## Symbols

-db <db>  
    reader command line option, 14  
-new-only, -no-new-only  
    reader-update command line option, 16  
-plugin <plugin>  
    reader command line option, 14  
-update, -no-update  
    reader-add command line option, 14  
-h, -host <host>  
    reader-serve command line option, 15  
-p, -port <port>  
    reader-serve command line option, 15  
-v, -verbose  
    reader-add command line option, 14  
    reader-remove command line option, 15  
    reader-serve command line option, 15  
    reader-update command line option, 16

## A

add\_feed() (reader.Reader method), 7  
author (reader.Entry attribute), 10  
author (reader.Feed attribute), 9

## C

Content (class in reader), 10  
content (reader.Entry attribute), 10

## E

Enclosure (class in reader), 10  
enclosures (reader.Entry attribute), 10  
Entry (class in reader), 9  
EntryError, 11  
EntryNotFoundError, 11

## F

Feed (class in reader), 9  
feed (reader.Entry attribute), 10  
FeedError, 10

FeedExistsError, 10  
FeedNotFoundError, 11

## G

get\_entries() (reader.Reader method), 8  
get\_feed() (reader.Reader method), 8  
get\_feeds() (reader.Reader method), 8

## H

href (reader.Enclosure attribute), 10

## I

id (reader.Entry attribute), 9  
id (reader.EntryError attribute), 11

## L

language (reader.Content attribute), 10  
length (reader.Enclosure attribute), 10  
link (reader.Entry attribute), 9  
link (reader.Feed attribute), 9

## M

mark\_as\_read() (reader.Reader method), 9  
mark\_as\_unread() (reader.Reader method), 9

## P

ParseError, 11  
published (reader.Entry attribute), 10

## R

read (reader.Entry attribute), 10  
Reader (class in reader), 7  
reader (module), 7, 23, 27  
reader command line option  
    -db <db>, 14  
    -plugin <plugin>, 14  
reader-add command line option  
    -update, -no-update, 14  
    -v, -verbose, 14

URL, 14

reader-remove command line option

-v, --verbose, 15

URL, 15

reader-serve command line option

-h, --host <host>, 15

-p, --port <port>, 15

-v, --verbose, 15

reader-update command line option

--new-only, --no-new-only, 16

-v, --verbose, 16

URL, 16

reader.plugins.feed\_entry\_dedupe (module), 24

reader.plugins.regex\_mark\_as\_read (module), 23

reader.plugins.tumblr\_gdpr (module), 24

ReaderError, 10

remove\_feed() (reader.Reader method), 7

## S

set\_feed\_user\_title() (reader.Reader method), 8

StorageError, 11

summary (reader.Entry attribute), 10

## T

title (reader.Entry attribute), 9

title (reader.Feed attribute), 9

type (reader.Content attribute), 10

type (reader.Enclosure attribute), 10

## U

update\_feed() (reader.Reader method), 8

update\_feeds() (reader.Reader method), 8

updated (reader.Entry attribute), 9

updated (reader.Feed attribute), 9

URL

reader-add command line option, 14

reader-remove command line option, 15

reader-update command line option, 16

url (reader.EntryError attribute), 11

url (reader.Feed attribute), 9

url (reader.FeedError attribute), 10

user\_title (reader.Feed attribute), 9

## V

value (reader.Content attribute), 10