
LDS Notes Documentation

Release 0.1.5

Easton Potokar

Mar 21, 2021

CONTENTS:

1	Roadmap	3
2	Done	5
3	Handling Highlights	7
4	Credits	9
4.1	Installation	9
4.2	Usage	10
4.3	API Reference	10
4.4	Contributing	14
4.5	History	16
4.6	Indices and tables	16
	Index	17

Unofficial Python API to interact with your annotations from churchofjesuschrist.org. I reverse engineered a bit of the API to download content/user notes from churchofjesuschrist.org. Currently can only download notes, working on uploading next.

- Free software: MIT license
- Documentation: <https://ldsnotes.readthedocs.io>.

ROADMAP

- Update Content/Annotation classes to inherit from addict.Dict. Should make upload easier later.
- 2-way sync.

DONE

- Make tests (and setup CI with travis) using dummy account/notes.

HANDLING HIGHLIGHTS

The way churchofjesuschrist.org handles where highlights are is a bit difficult to reverse engineer. They save where your highlight is by counting words - both from the start, and from the end. The difficult part is figuring out what they consider a “word”. For example, a footnote/reference counts as a word, *and a comma after a word with a footnote also counts as one*. This makes things very case by case to get things right. If you have a problem with a highlight being a few words off, please open an issue with your where your highlight is at.

TL;DR Highlights are hard, open issue if yours are off.

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

4.1 Installation

4.1.1 Stable release

To install LDS Notes, run this command in your terminal:

```
$ pip install ldsnotes
```

This is the preferred method to install LDS Notes, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

4.1.2 From sources

The sources for LDS Notes can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/contagon/ldsnotes
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/contagon/ldsnotes/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

4.2 Usage

Basic note grabbing would go something like this:

```
from pprint import pprint
from ldsnotes import Notes

n = Notes("username", "password")

pprint(n[:10])
```

LDS Notes using selenium to login in to churchofjesuschrist.org. Basically a browser opens in the background and logs you in. This can take ~5 seconds and is slow to do everytime you need to run your script. Alternatively, you can save your token and reuse it (they seem to be good for a couple of hours).

I generally do the best of both worlds by saving my token to a file locally:

```
try:
    file = open(".token", 'r')
    token = file.read()
    file.close()
    n = Notes(token=token)
    # try to grab something with token
    n[0]
except:
    print("Refetching token...")
    n = Notes("username", "password")
    file = open(".token", "w")
    file.write(n.token)
    file.close()
```

You can also search specifically for what you need:

```
notes = n.search(keyword="hope", folder="Studying", annot_type="reference", start=1,
↳ stop=100)
```

See *API Reference* for more specifics.

4.3 API Reference

4.3.1 Notes

class ldsnotes.Notes (username=None, password=None, token=None, headless=True)

Wrapper to pull any annotations from lds.org.

The API is rather complex to use to login. We take the lazy route and login with selenium (basically a fake browser). To avoid doing this everytime you can save the necessary token.

The object also supports indexing, so you can get the first note with `n[0]`, or the first 10 doing `n[:10]`. When doing this it'll return the most recently edited objects. Whenever querying, it'll always return the most recently edited objects.

Parameters

- **username** (*string*) – Your username
- **password** (*string*) – Your password

- **token** (*string*) – Instead of inputting your username/password, you can save your token and just input it
- **headless** (*bool*) – Whether to run selenium headless or not

Variables

- **tags** (*list*) – List of Tag objects of all your tags
- **folders** (*list*) – List of Folder objects of all your folders

class ldsnotes.**Tag** (**args, **kwargs*)

Object that holds all Tag info

Variables

- **annotationCount** (*string*) – Number of annotations assigned to tag
- **name** (*string*) – Name of tag
- **id** (*string*) – Seems to always be the same as name...
- **lastUsed** (*datetime*) – Last time that tag was edited

class ldsnotes.**Folder** (**args, **kwargs*)

Object that holds all Tag info

Variables

- **annotationCount** (*string*) – Number of annotations in folder
- **name** (*string*) – Name of folder
- **id** (*string*) – Special id of folder
- **lastUsed** (*datetime*) – Last time that the folder was changed
- **order.id** (*list*) – List of note ids in order that they were put in.

4.3.2 Annotations

class ldsnotes.**Annotation** (*json*)

Base class for all annotations.

Variables

- **tags** (*list*) – List of strings of tag names.
- **folders_id** (*list*) – List of strings of folder ids.
- **last_update** (*datetime*) – Last time annotation was edited.
- **id** (*string*) – Id of annotation.

class ldsnotes.**Bookmark** (*json*)

A bookmark annotation. Inherits from annotation.

Variables

- **tags** (*list*) – List of strings of tag names.
- **folders_id** (*list*) – List of strings of folder ids.
- **last_update** (*datetime*) – Last time annotation was edited.
- **id** (*string*) – Id of annotation.
- **headline** (*string*) – Name of article ie name of conference talk or Helaman 3.

- **reference** (*string*) – Full reference for scriptures like Helaman 3:29.
- **publication** (*string*) – Refers to book (ie GC 2020 or BoM).
- **url** (*string*) – Url to bookmark location.

class `ldsnotes.Journal` (*json*)

A journal entry. Inherits from Annotation.

Variables

- **tags** (*list*) – List of strings of tag names.
- **folders_id** (*list*) – List of strings of folder ids.
- **last_update** (*datetime*) – Last time annotation was edited.
- **id** (*string*) – Id of annotation.
- **title** (*string*) – Title of journal entry.
- **note** (*string*) – Actual note taken.

class `ldsnotes.Highlight` (*json, content_jsons*)

A scripture highlight. Inherits from Journal.

Variables

- **tags** (*list*) – List of strings of tag names.
- **folders_id** (*list*) – List of strings of folder ids.
- **last_update** (*datetime*) – Last time annotation was edited.
- **id** (*string*) – Id of annotation.
- **title** (*string*) – Title of journal entry.
- **note** (*string*) – Actual note taken.
- **color** (*string*) – Color of annotation.
- **content** (*string*) – Content of verse/paragraph(s) being highlighted.
- **hl** (*string*) – Portion of verse that's been highlighted.
- **url** (*string*) – Url to verse/paragraph(s)
- **headline** (*string*) – Name of article ie name of conference talk or Helaman 3.
- **reference** (*string*) – Full reference for scriptures like Helaman 3:29.
- **publication** (*string*) – Refers to book (ie GC 2020 or BoM).

markdown (*syntax*='==')

Returns content with the highlight wrapped in markdown syntax.

Parameters **syntax** (*string*) – String to wrap highlight in. Defaults to ==

Returns

Return type Content with wrapped highlight.

class `ldsnotes.Reference` (*json, hl_json, ref_json*)

A scripture link/reference. Inherits from Highlight.

Variables

- **tags** (*list*) – List of strings of tag names.

- **folders_id** (*list*) – List of strings of folder ids.
- **last_update** (*datetime*) – Last time annotation was edited.
- **id** (*string*) – Id of annotation.
- **title** (*string*) – Title of journal entry.
- **note** (*string*) – Actual note taken.
- **color** (*string*) – Color of annotation.
- **content** (*string*) – Content of verse/paragraph(s) being highlighted.
- **hl** (*string*) – Portion of verse that's been highlighted.
- **url** (*string*) – Url to verse/paragraph(s)
- **headline** (*string*) – Name of article ie name of conference talk or Helaman 3.
- **reference** (*string*) – Full reference for scriptures like Helaman 3:29.
- **publication** (*string*) – Refers to book (ie GC 2020 or BoM).
- **ref_content** (*string*) – Content of verse/paragraph(s) that are linked to
- **ref_headline** (*string*) – Headline of reference. See headline for examples.
- **ref_reference** (*string*) – Reference of reference. See reference for examples.
- **ref_publication** (*string*) – Publication of reference. See publication for examples.

markdown (*syntax*='==')

Returns content with the highlight wrapped in markdown syntax.

Parameters **syntax** (*string*) – String to wrap highlight in. Defaults to ==

Returns

Return type Content with wrapped highlight.

4.3.3 Content

class `ldsnotes.Content` (*json*)

Class that pulls/represents content from anywhere on churchofjesuschrist.org/study (theoretically)

Parameters **json** (*dict*) – Dictionary made from json pull from lds.org's API.

Variables

- **content** (*string*) – Book, talk, or section of content.
- **headline** (*string*) – The content (see above) with verse number in case of scriptures.
- **publication** (*string*) – Overarching publication. Think BoM, DoC, General Conference 2020, etc.
- **url** (*string*) – URL of where the content is located (including the paragraph/verse).
- **uri** (*string*) – URI that it was pulled with.
- **p_start** (*string*) – First verse/paragraph pulled.
- **p_end** (*string*) – Last verse/paragraph pulled.

4.4 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.4.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/contagon/ldsnotes/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

LDS Notes could always use more documentation, whether as part of the official LDS Notes docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/contagon/ldsnotes/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.4.2 Get Started!

Ready to contribute? Here's how to set up *ldsnotes* for local development.

1. Fork the *ldsnotes* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ldsnotes.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ldsnotes
$ cd ldsnotes/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 ldsnotes tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.7, 3.8, 3.9 and for PyPy. Check https://travis-ci.com/contagon/ldsnotes/pull_requests and make sure that the tests pass for all supported Python versions.

4.4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_ldsnotes
```

4.4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

4.5 History

4.5.1 0.1.0 (2020-24-20)

- Wrote tests to check basic functionality
- Got travis deployment encryption set up
- Various small bug fixes found when making tests

4.5.2 0.1.0 (2020-22-20)

- First release on PyPI.

4.6 Indices and tables

- genindex
- modindex
- search

INDEX

A

Annotation (*class in ldsnotes*), 11

B

Bookmark (*class in ldsnotes*), 11

C

Content (*class in ldsnotes*), 13

F

Folder (*class in ldsnotes*), 11

H

Highlight (*class in ldsnotes*), 12

J

Journal (*class in ldsnotes*), 12

M

markdown() (*ldsnotes.Highlight method*), 12

markdown() (*ldsnotes.Reference method*), 13

N

Notes (*class in ldsnotes*), 10

R

Reference (*class in ldsnotes*), 12

T

Tag (*class in ldsnotes*), 11