
pyliveleak Documentation

Release 0.1.2

Michael Penkov

Aug 10, 2017

Contents

1	pyliveleak	3
1.1	Features	3
1.2	Credits	4
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
4	pyliveleak	9
4.1	pyliveleak package	9
5	Contributing	13
5.1	Types of Contributions	13
5.2	Get Started!	14
5.3	Pull Request Guidelines	15
5.4	Tips	15
6	Credits	17
6.1	Development Lead	17
6.2	Contributors	17
7	History	19
7.1	0.1.2 (2017-08-11)	19
7.2	0.1.1 (2017-08-10)	19
7.3	0.1.0 (2017-08-10)	19
8	Indices and tables	21
	Python Module Index	23

Contents:

Uploads videos to liveleak.com

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

1.1 Features

Sample usage:

```
$ pyliveleak --path tests/test-data/foreman_cif.mp4 --username "$username" --password  
↪ "$password"  
https://www.liveleak.com/view?i=7ed_1502358506
```

For additional options:

```
$ pyliveleak --help  
Usage: pyliveleak [OPTIONS]  
  
    Console script for pyliveleak.  
  
Options:  
  --loglevel INTEGER      Your liveleak.com loglevel [required]  
  --password TEXT         Your liveleak.com password [required]  
  --username TEXT         Your liveleak.com username [required]  
  --path PATH             The video to upload [required]  
  --help                  Show this message and exit.
```

You may also use pyliveleak as a Python library:

```
>>> import pyliveleak  
>>> index_page = pyliveleak.login(username, password)  
>>> file_token, item_token = index_page.add_item('tests/test-data/foreman_cif.mp4')
```

```
>>> item_token
u'b86_1502357642'
```

Your new video will be available here: https://www.liveleak.com/view?i={item_token}

You may specify optional metadata:

```
>>> index_page.add_item('tests/test-data/foreman_cif.mp4', title='my title',
...                    body='detailed description', tags='tags', category='World News
↳')
```

The category must be one of:

```
>>> pprint.pprint(sorted(pyliveleak.CATEGORIES))
['afghanistan',
 'citizen journalism',
 'conspiracy',
 'creative',
 'history',
 'hobbies',
 'iran',
 'iraq',
 'liveleak challenges',
 'liveleaks',
 'music',
 'nature',
 'other',
 'other entertainment',
 'other items from liveleakers',
 'other middle east',
 'other news',
 'politics',
 'propaganda',
 'regional news',
 'religion',
 'science and technology',
 'sports',
 'syria',
 'ukraine',
 'vehicles',
 'weapons',
 'world news',
 'wtf',
 'yawn',
 'your say']
```

1.2 Credits

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

2.1 Stable release

To install pyliveleak, run this command in your terminal:

```
$ pip install pyliveleak
```

This is the preferred method to install pyliveleak, 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.

2.2 From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/mpenkov/pyliveleak
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/mpenkov/pyliveleak/tarball/master
```

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

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


CHAPTER 3

Usage

To use pyliveleak in a project:

```
import pyliveleak
```


4.1 pyliveleak package

4.1.1 Submodules

4.1.2 pyliveleak.cli module

Console script for pyliveleak.

4.1.3 pyliveleak.pyliveleak module

Main module.

```
class pyliveleak.pyliveleak.AddItemPage (html, cookies)  
    Bases: object
```

This internal class performs the hard work for adding a video to liveleak.com.

That is a three-stage process:

1. Upload the video file to liveleak's AWS S3 bucket.
2. Add the file to liveleak, yielding a file token.
3. Specify metadata and publish the video.

```
__init__ (html, cookies)
```

```
add_file (path, aws_response)
```

```
connect_string
```

Parse the connect_string from the page.

This is unique for each page load.

connection

Parse the connection number from the page.

This is unique for each page load.

multipart_params

Parse the multipart_params dict from the JavaScript in the page.

We need these params to upload the file to AWS.

publish (*title='title', body='body', tags='liveleak.py', category='other'*)

upload_to_aws (*path*)

Upload a file to AWS. Raises Exception on failure. :return: a file_token in case of success :rtype: str

class pyliveleak.pyliveleak.**IndexPage** (*html, cookies*)

Bases: object

Represents a logged-in session with liveleak.com.

To obtain an instance of this class, see the `pyliveleak.login()` function.

__init__ (*html, cookies*)

Internal constructor.

Parameters

- **html** (*str*) – The HTML from the index page.
- **cookies** (*str*) – A dictionary of cookies containing the entire session.

add_item (*path, title=None, body=None, tags='liveleak.py', category='other'*)

Upload a video to liveleak.com.

Parameters

- **path** (*str*) – The full path to the video to upload.
- **title** (*str*) – The title of the video. If None, will be the filename of the video.
- **description** (*str*) – A description of the video. If None, will be the filename of the video.
- **tags** (*str*) – Tags for the video.
- **category** (*str*) – The name of the category for the video.

Returns The file_token and item_token as a tuple.

You may use the item_token to access the uploaded video through your browser:: https://www.liveleak.com/view?i={item_token}

exception pyliveleak.pyliveleak.**PyLiveleakException**

Bases: exceptions.RuntimeError

Raised when something unexpected happens within pyliveleak.

`pyliveleak.pyliveleak.load_categories()`

`pyliveleak.pyliveleak.login(username, password)`

Login to liveleak.com.

This is the main entry point to pyliveleak. Once you login, you can upload videos.

Parameters

- **username** (*str*) – Your liveleak.com username

- **password** (*str*) – Your liveleak.com password

Returns A new session with liveleak.com.

Return type *pyliveleak.IndexPage*

4.1.4 Module contents

Sample usage:

```
>>> import pyliveleak
>>> index_page = pyliveleak.login(username, password)
>>> file_token, item_token = index_page.add_item('tests/test-data/foreman_cif.mp4')
>>> item_token
u'b86_1502357642'
```

Your new video will be available here: https://www.liveleak.com/view?i={item_token}

You may specify optional metadata:

```
>>> index_page.add_item('tests/test-data/foreman_cif.mp4', title='my title',
...                     body='detailed description', tags='tags', category='World News
↪')
```

The category must be one of:

```
>>> pprint.pprint(sorted(pyliveleak.CATEGORIES))
['afghanistan',
 'citizen journalism',
 'conspiracy',
 'creative',
 'history',
 'hobbies',
 'iran',
 'iraq',
 'liveleak challenges',
 'liveleaks',
 'music',
 'nature',
 'other',
 'other entertainment',
 'other items from liveleakers',
 'other middle east',
 'other news',
 'politics',
 'propaganda',
 'regional news',
 'religion',
 'science and technology',
 'sports',
 'syria',
 'ukraine',
 'vehicles',
 'weapons',
 'world news',
 'wtf',
 'yawn',
 'your say']
```

`pyliveleak.login(username, password)`

Login to liveleak.com.

This is the main entry point to pyliveleak. Once you login, you can upload videos.

Parameters

- **username** (*str*) – Your liveleak.com username
- **password** (*str*) – Your liveleak.com password

Returns A new session with liveleak.com.

Return type `pyliveleak.IndexPage`

class `pyliveleak.IndexPage(html, cookies)`

Bases: `object`

Represents a logged-in session with liveleak.com.

To obtain an instance of this class, see the `pyliveleak.login()` function.

__init__ (*html, cookies*)

Internal constructor.

Parameters

- **html** (*str*) – The HTML from the index page.
- **cookies** (*str*) – A dictionary of cookies containing the entire session.

add_item (*path, title=None, body=None, tags='liveleak.py', category='other'*)

Upload a video to liveleak.com.

Parameters

- **path** (*str*) – The full path to the video to upload.
- **title** (*str*) – The title of the video. If None, will be the filename of the video.
- **description** (*str*) – A description of the video. If None, will be the filename of the video.
- **tags** (*str*) – Tags for the video.
- **category** (*str*) – The name of the category for the video.

Returns The `file_token` and `item_token` as a tuple.

You may use the `item_token` to access the uploaded video through your browser:: https://www.liveleak.com/view?i={item_token}

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/mpenkov/pyliveleak/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mpenkov/pylivelack/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 :)

5.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pylivelack.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 pylivelack
$ cd pylivelack/
$ 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 pylivelack tests
$ python setup.py test or py.test
$ 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.

5.3 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/mpenkov/pyliveleak/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_pyliveleak
```


CHAPTER 6

Credits

6.1 Development Lead

- Michael Penkov <misha.penkov@gmail.com>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.2 (2017-08-11)

7.1.1 Fixed

- Include categories.yml (and VERSION) in package data, for real this time.

7.2 0.1.1 (2017-08-10)

7.2.1 Added

- More Sphinx documentation

7.2.2 Fixed

- Include categories.yml in package data.
- Update Sphinx version so that documentation builds properly.

7.3 0.1.0 (2017-08-10)

- First release on PyPI.

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pyliveleak`, [11](#)

`pyliveleak.cli`, [9](#)

`pyliveleak.pyliveleak`, [9](#)

Symbols

`__init__()` (pylivelack.IndexPage method), [12](#)
`__init__()` (pylivelack.pylivelack.AddItemPage method), [9](#)
`__init__()` (pylivelack.pylivelack.IndexPage method), [10](#)

A

`add_file()` (pylivelack.pylivelack.AddItemPage method), [9](#)
`add_item()` (pylivelack.IndexPage method), [12](#)
`add_item()` (pylivelack.pylivelack.IndexPage method), [10](#)
`AddItemPage` (class in pylivelack.pylivelack), [9](#)

C

`connect_string` (pylivelack.pylivelack.AddItemPage attribute), [9](#)
`connection` (pylivelack.pylivelack.AddItemPage attribute), [9](#)

I

`IndexPage` (class in pylivelack), [12](#)
`IndexPage` (class in pylivelack.pylivelack), [10](#)

L

`load_categories()` (in module pylivelack.pylivelack), [10](#)
`login()` (in module pylivelack), [11](#)
`login()` (in module pylivelack.pylivelack), [10](#)

M

`multipart_params` (pylivelack.pylivelack.AddItemPage attribute), [10](#)

P

`publish()` (pylivelack.pylivelack.AddItemPage method), [10](#)
`pylivelack` (module), [11](#)
`pylivelack.cli` (module), [9](#)
`pylivelack.pylivelack` (module), [9](#)
`PyLiveleakException`, [10](#)

U

`upload_to_aws()` (pylivelack.pylivelack.AddItemPage method), [10](#)