Lassie Documentation

Release 0.6.2

Mike Helmick

November 11, 2015

Contents

Lassie is a Python library for retrieving basic content from websites.

CHAPTER 1

Usage

```
>>> import lassie
>>> lassie.fetch('http://www.youtube.com/watch?v=dQw4w9WgXcQ')
{
    'description': u'Music video by Rick Astley performing Never Gonna Give You Up. YouTube view cour
    'videos': [{
        'src': u'http://www.youtube.com/v/dQw4w9WgXcQ?autohide=1&version=3',
        'height': 480,
        'type': u'application/x-shockwave-flash',
        'width': 640
    }, {
        'src': u'https://www.youtube.com/embed/dQw4w9WgXcQ',
        'height': 480,
        'width': 640
    }],
    'title': u'Rick Astley - Never Gonna Give You Up',
    'url': u'http://www.youtube.com/watch?v=dQw4w9WgXcQ',
    'keywords': [u'Rick', u'Astley', u'Sony', u'BMG', u'Music', u'UK', u'Pop'],
    'images': [{
        'src': u'http://i1.ytimg.com/vi/dQw4w9WgXcQ/hqdefault.jpg?feature=og',
        'type': u'og:image'
    }, {
        'src': u'http://i1.ytimg.com/vi/dQw4w9WgXcQ/hqdefault.jpg',
        'type': u'twitter:image'
    }, {
        'src': u'http://s.ytimg.com/yts/img/favicon-vfldLzJxy.ico',
        'type': u'favicon'
    }, {
        'src': u'http://s.ytimg.com/yts/img/favicon_32-vflWoMFGx.png',
        'type': u'favicon'
    }],
    'locale': u'en_US'
}
```

CHAPTER 2

User Guide

2.1 Installation

Information on how to properly install Lassie

2.1.1 Pip or Easy Install

Install Lassie via pip

```
$ pip install lassie
```

or, with easy_install

```
$ easy_install lassie
```

But, hey... that's up to you.

2.1.2 Source Code

Lassie is actively maintained on GitHub

Feel free to clone the repository

```
git clone git://github.com/michaelhelmick/lassie.git
```

tarball

```
$ curl -OL https://github.com/michaelhelmick/lassie/tarball/master
```

zipball

```
$ curl -OL https://github.com/michaelhelmick/lassie/zipball/master
```

Now that you have the source code, install it into your site-packages directory

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

So Lassie is installed! Now, head over to the *starting out* section.

2.2 Starting Out

This section out lines the most basic uses of Lassie

2.2.1 What Lassie Returns

Lassie aims to return the most beautifully crafted dictionary of important information about the web page.

2.2.2 Beginning

So, let's say you want to retrieve details about a YouTube video.

Specifically: http://www.youtube.com/watch?v=dQw4w9WgXcQ

```
>>> import lassie
>>> lassie.fetch('http://www.youtube.com/watch?v=dQw4w9WgXcQ')
{
    'description': u'Music video by Rick Astley performing Never Gonna Give You Up. YouTube view cou
    'videos': [{
        'src': u'http://www.youtube.com/v/dQw4w9WgXcQ?version=3&autohide=1',
        'height': 480,
        'type': u'application/x-shockwave-flash',
        'width': 640
    }, {
        'src': u'https://www.youtube.com/embed/dQw4w9WgXcQ',
        'height': 480,
        'width': 640
    }],
    'title': u'Rick Astley - Never Gonna Give You Up',
    'url': u'http://www.youtube.com/watch?v=dQw4w9WgXcQ',
    'keywords': [u'Rick', u' Astley', u' Sony', u' BMG', u' Music', u' UK', u' Pop'],
    'images': [{
        'src': u'http://i1.ytimg.com/vi/dQw4w9WgXcQ/hqdefault.jpg?feature=og',
        'type': u'og:image'
    }, {
        'src': u'http://i1.ytimg.com/vi/dQw4w9WgXcQ/hqdefault.jpg',
        'type': u'twitter:image'
    }, {
        'src': u'http://s.ytimg.com/yts/img/favicon-vfldLzJxy.ico',
        'type': u'favicon'
    }, {
        'src': u'http://s.ytimg.com/yts/img/favicon_32-vflWoMFGx.png',
        'type': u'favicon'
    }],
    'locale': u'en_US'
}
```

Or what if you wanted to get information about an article?

Specifically: http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/

```
>>> import lassie
>>> lassie.fetch('http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/')
{
    'description': u"GitHub has surpassed the 3 million-developer mark, a milestone for the collabora
```

```
    'videos': [],
    'title': u'GitHub Passes The 3 Million Developer Mark | TechCrunch',
    'url': u'http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/',
    'locale': u'en_US',
    'images': [{
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png?w=150',
        'type': u'og:image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png',
        'type': u'twitter:image'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }]
}
```

Lassie, by default, also filters for content from Twitter Cards, grab favicons and touch icons.

2.2.3 Priorities

Open Graph values takes priority over other values (Twitter Card data, generic data, etc.)

In other words, if a website has the title of their page as `<title>YouTube</title>` and they have their Open Graph title set `<meta property="og:title" content="YouTube | A Video Sharing Site" />`

The value of `title` when you `fetch` the web page will return as "YouTube | A Video Sharing Site" instead of just "YouTube".

But what if I don't want open graph data?

Then pass `open_graph=False` to the `fetch` method.

```
>>> lassie.fetch('http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/', oper
{
    'description': u"GitHub has surpassed the 3 million-developer mark, a milestone for the collabora
    'videos': [],
    'title': u'GitHub Passes The 3 Million Developer Mark | TechCrunch',
    'url': u'http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/',
    'locale': u'en_US',
    'images': [{
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png?w=150',
        'type': u'og:image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png',
        'type': u'twitter:image'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }]
}
```

If you **don't** want Twitter cards, favicons or touch icons, use any combination of the following parameters and pass them to `fetch`:

- Pass `twitter_card=False` to exclude Twitter Card data from being filtered
- Pass `touch_icon=False` to exclude the Apple touch icons from being added to the images array
- Pass `favicon=False` to exclude the favicon from being added to the images array

2.2.4 Obtaining All Images

Sometimes you might want to obtain a list of all the images on a web page... simple, just pass `all_images=True` to `fetch`.

```
>>> lassie.fetch('http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/', all_
{
    'description': u"GitHub has surpassed the 3 million-developer mark, a milestone for the collabora
    'videos': [],
    'title': u'GitHub Passes The 3 Million Developer Mark | TechCrunch',
    'url': u'http://techcrunch.com/2013/01/16/github-passes-the-3-million-developer-mark/',
    'locale': u'en_US',
    'images': [{
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png?w=150',
        'type': u'og:image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png',
        'type': u'twitter:image'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/favicon.ico?m=135766010
        'type': u'favicon'
    }, {
        'src': u'http://s2.wp.com/wp-content/themes/vip/tctechcrunch2/images/site-logo-cutout.png?m=1
        'alt': u'',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/countdown4.jpg?w=640',
        'alt': u'Main Event Page',
        'type': u'body_image'
    }, {
        'src': u'http://2.gravatar.com/avatar/b4e205744ae2f9b44921d103b4d80e54?s=60&d=identicon&r=G',
        'alt': u'',
        'height': 60,
        'type': u'body_image',
        'width': 60
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/01/github-logo.png?w=300',
        'alt': u'github-logo',
        'height': 300,
        'type': u'body_image',
        'width': 300
    }, {
        'src': u'http://crunchbase.com/assets/images/resized/0001/7208/17208v9-max-150x150.png',
        'alt': u'',
        'type': u'body_image'
    }, {
```

```
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/tardis-egg.jpg?w=89&h=64&crop=1
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/made-in-space-zero-gravity.jpg?w
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/04/apple1.jpg?w=89&h=64&crop=1',
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/p9130014.jpg?w=89&h=64&crop=1',
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/htc.png?w=89&h=64&crop=1',
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/screen-shot-2013-08-13-at-8-18-2
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/24112v5-max-250x250.jpg?w=89&h=6
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/surface-14.jpg?w=89&h=64&crop=1
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/sprawl_tuned_robot.jpg?w=89&h=64
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/ashton-kutcher-jobs.jpg?w=89&h=6
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/facebook-commerce.png?w=89&h=648
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/screen-shot-2013-08-14-at-10-23-
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2012/10/ibm_logo.jpg?w=89&h=64&crop=1',
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/screen-shot-2013-08-15-at-12-09-
        'alt': '',
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/inklogo.jpg?w=89&h=64&crop=1',
        'alt': '',
```

```
        'type': u'body_image'
    }, {
        'src': u'http://tctechcrunch2011.files.wordpress.com/2013/08/screen-shot-2013-08-15-at-9-31-2
        'alt': '',
        'type': u'body_image'
    }]
}
```

So, now you know the basics. What if you don't want to declare params *every* time to the `fetch` method? Head over to the *advanced usage* section to learn about the `Lassie` class.

2.3 Advanced Usage

This section will cover how to use the `Lassie` class to maintain settings across all `fetch` calls.

2.3.1 Class Level Attributes

Constructing a `Lassie` class and calling `fetch` will use all the default params that are available to `fetch`.

```
>>> from lassie import Lassie
>>> l = Lassie()

>>> l.fetch('https://github.com/michaelhelmick')
{
    'images': [{
        'src': u'https://github.global.ssl.fastly.net/images/modules/logos_page/Octocat.png',
        'type': u'og:image'
    }, {
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/michaelhelmick',
    'description': u'michaelhelmick has 22 repositories written in Python, Shell, and JavaScript. Fo
    'videos': [],
    'title': u'michaelhelmick (Mike Helmick) \xb7 GitHub'
}
>>> l.fetch('https://github.com/ashibble')
{
    'images': [{
        'src': u'https://github.global.ssl.fastly.net/images/modules/logos_page/Octocat.png',
        'type': u'og:image'
    }, {
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/ashibble',
    'description': u'Follow ashibble on GitHub and watch them build beautiful projects.',
    'videos': [],
    'title': u'ashibble (Alexander Shibble) \xb7 GitHub'
}
```

If you decide that you don't want to filter for Open Graph data, instead of declaring `open_graph=False` in every `fetch` call:

```
>>> import lassie
>>> l = Lassie()
>>> l.fetch('https://github.com/michaelhelmick', open_graph=False)
>>> l.fetch('https://github.com/ashibble', open_graph=False)
```

You can use the `Lassie` class and set attibutes on the class.

```
>>> from lassie import Lassie
>>> l = Lassie()
>>> l.open_graph = False

>>> l.fetch('https://github.com/michaelhelmick')
{
    'images': [{
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/michaelhelmick',
    'description': u'michaelhelmick has 22 repositories written in Python, Shell, and JavaScript. Fo
    'videos': [],
    'title': u'michaelhelmick (Mike Helmick) \xb7 GitHub'
}
>>> l.fetch('https://github.com/ashibble')
{
    'images': [{
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/ashibble',
    'description': u'Follow ashibble on GitHub and watch them build beautiful projects.',
    'videos': [],
    'title': u'ashibble (Alexander Shibble) \xb7 GitHub'
}
```

You'll notice the data for the Open Graph properties wasn't returned in the last responses. That's because passing `open_graph=False` tells Lassie to not filter for those properties.

In the edge case that there is a time or two you want to override the class attribute, just pass the parameter to `fetch` and Lassie will use that parameter.

```
>>> from lassie import Lassie
>>> l = Lassie()
>>> l.open_graph = False

>>> l.fetch('https://github.com/michaelhelmick')
{
    'images': [{
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/michaelhelmick',
    'description': u'michaelhelmick has 22 repositories written in Python, Shell, and JavaScript. Fo
    'videos': [],
    'title': u'michaelhelmick (Mike Helmick) \xb7 GitHub'
}
>>> l.fetch('https://github.com/ashibble', open_graph=True)
{
    'images': [{
        'src': u'https://github.global.ssl.fastly.net/images/modules/logos_page/Octocat.png',
```

```
        'type': u'og:image'
    }, {
        'src': u'https://github.com/favicon.ico',
        'type': u'favicon'
    }],
    'url': 'https://github.com/ashibble',
    'description': u'Follow ashibble on GitHub and watch them build beautiful projects.',
    'videos': [],
    'title': u'ashibble (Alexander Shibble) \xb7 GitHub'
}
```

2.3.2 Manipulate the Request (headers, proxies, etc.)

There are times when you may want to turn SSL verification off, send custom headers, or add proxies for the request to go through.

Lassie uses the requests library to make web requests. `requests` accepts a few parameters to allow developers to manipulate the acutal HTTP request.

Here is an example of sending custom headers to a lassie request:

```
from lassie import Lassie

l = Lassie()
l.request_opts = {
    'headers': {
        'User-Agent': 'python lassie'
    }
}
l.fetch('http://google.com')
```

Maybe you want to set a request timeout, here's another example:

```
from lassie import Lassie

l = Lassie()
l.request_opts = {
    'timeout': 10  # 10 seconds
}

# If the response takes longer than 10 seconds this request will fail
l.fetch('http://google.com')
```

2.3.3 Playing Nice with non-HTML Files

Sometimes, you may want to grab information about an image or other type of file. Although only images are supported, you can retrieve a nicely structured `dict`

Pass `handle_file_content=True` to `lassie.fetch` or set it on a `Lassie` instance

```
>>> from lassie import Lassie

>>> lassie.fetch('https://camo.githubusercontent.com/d19b279de191489445d8cfd39faf93e19ca2df14/687474
{
    'title': '68747470733a2f2f692e696d6775722e636f6d2f5172764e6641582e676966',
    'videos': [],
    'url': 'https://camo.githubusercontent.com/d19b279de191489445d8cfd39faf93e19ca2df14/68747470733a2
```

```
    'images': [{
        'type': 'body_image',
        'src': 'https://camo.githubusercontent.com/d19b279de191489445d8cfd39faf93e19ca2df14/68747470
    }]
}

>>> lassie.fetch('http://2.bp.blogspot.com/-vzGgFFtW-VY/Tz-eozaHw3I/AAAAAAAAM3k/OMvxpFYr23s/s1600/The
{
    'title': 'The-best-top-desktop-cat-wallpapers-10.jpg',
    'images': [{
        'type': 'body_image',
        'src': 'http://2.bp.blogspot.com/-vzGgFFtW-VY/Tz-eozaHw3I/AAAAAAAAM3k/OMvxpFYr23s/s1600/The-b
    }],
    'videos': [],
    'url': 'http://2.bp.blogspot.com/-vzGgFFtW-VY/Tz-eozaHw3I/AAAAAAAAM3k/OMvxpFYr23s/s1600/The-best-
}
```

CHAPTER 3

Lassie API Documentation

3.1 Developer Interface

This page of the documentation will cover all methods and classes available to the developer.

3.1.1 Core Interface

class `lassie.`**`Lassie`**

`__init__`()
Instantiates an instance of Lassie.

`fetch`(*url*, *open_graph=None*, *twitter_card=None*, *touch_icon=None*, *favicon=None*, *all_images=None*, *parser=None*, *handle_file_content=None*, *canonical=None*)
Retrieves content from the specified url, parses it, and returns a beautifully crafted dictionary of important information about that web page.

Priority tree is as follows:

1. Open Graph
2. Twitter Card
3. Other meta content (i.e. description, keywords)

Parameters

- **`url`** – URL to send a GET request to
- **`open_graph`** (*bool*) – (optional) If `True`, filters web page content for Open Graph meta tags. The content of these properties have top priority on return values.
- **`twitter_card`** (*bool*) – (optional) If `True`, filters web page content for Twitter Card meta tags
- **`touch_icon`** (*bool*) – (optional) If `True`, retrieves Apple touch icons and includes them in the response `images` array
- **`favicon`** (*bool*) – (optional) If `True`, retrieves any favicon images and includes them in the response `images` array
- **`canonical`** (*bool*) – (optional) If `True`, retrieves canonical url from meta tags. Default: False

- **all_images** (*bool*) – (optional) If `True`, retrieves images inside web pages body and includes them in the response `images` array. Default: False
- **parser** (*string*) – (optional) String reference for the parser that BeautifulSoup will use
- **handle_file_content** (*bool*) – (optional) If `True`, lassie will return a generic response when a file is fetched. Default: False

3.1.2 Exceptions

exception `lassie.`**LassieError**
Generic catch-all Exceptions

l

Symbols

F

L