
pyPodcastParser Documentation

Release 0.1.0

Jason Rigden

December 10, 2016

1	Introduction	1
2	Installation	3
3	Usage	5
3.1	Objects and their Useful Attributes	5
4	Bugs & Feature Requests	9
5	Credits	11
5.1	Jason Rigden	11
6	Development	13
7	Testing	15
8	License	17
8.1	API	17

Introduction

pyPodcastParser is a podcast parser. It should parse any RSS file, but it specializes in parsing podcast rss feeds. pyPodcastParser is agnostic about the method you use to get a podcast RSS feed. Most user will be most comfortable with the [Requests](#) library.

Installation

```
pip install pyPodcastParser
```

Usage

```
from pyPodcastParser.Podcast import Podcast
import requests

request = requests.get('https://some_rss_feed')
podcast = Podcast(request.content)
```

3.1 Objects and their Useful Attributes

Notes:

- All attributes with empty or nonexistent element will have a value of None.
- Attributes are generally strings or lists of strings, because we want to record the literal value of elements.
- The cloud element aka RSS Cloud is not supported as it has been superseded by the superior PubSubHubbub protocol

3.1.1 Podcast

- categories (list) A list for strings representing the feed categories
- copyright (string): The feed's copyright
- creative_commons (string): The feed's creative commons license
- items (list): A list of Item objects
- description (string): The feed's description
- generator (string): The feed's generator
- image_title (string): Feed image title
- image_url (string): Feed image url
- image_link (string): Feed image link to homepage
- image_width (string): Feed image width
- image_height (Sample H4string): Feed image height
- itunes_author_name (string): The podcast's author name for iTunes
- itunes_block (boolean): Does the podcast block itunes

- itunes_categories (list): List of strings of itunes categories
- itunes_complete (string): Is this podcast done and complete
- itunes_explicit (string): Is this item explicit. Should only be “yes” and “clean.”
- itune_image (string): URL to itunes image
- itunes_keywords (list): List of strings of itunes keywords
- itunes_new_feed_url (string): The new url of this podcast
- language (string): Language of feed
- last_build_date (string): Last build date of this feed
- link (string): URL to homepage
- managing_editor (string): managing editor of feed
- published_date (string): Date feed was published
- pubsubhubbub (string): The URL of the pubsubhubbub service for this feed
- owner_name (string): Name of feed owner
- owner_email (string): Email of feed owner
- subtitle (string): The feed subtitle
- title (string): The feed title
- ttl (string): The time to live or number of minutes to cache feed
- web_master (string): The feed’s webmaster

3.1.2 Item

- author (string): The author of the item
- comments (string): URL of comments
- creative_commons (string): creative commons license for this item
- description (string): Description of the item.
- enclosure_url (string): URL of enclosure
- enclosure_type (string): File MIME type
- enclosure_length (integer): File size in bytes
- guid (string): globally unique identifier
- itunes_author_name (string): Author name given to iTunes
- itunes_block (boolean): It this Item blocked from itunes
- itunes_closed_captioned: (string): It is this item have closed captions
- itunes_duration (string): Duration of enclosure
- itunes_explicit (string): Is this item explicit. Should only be “yes” and “clean.”
- itune_image (string): URL of item cover art
- itunes_order (string): Override published_date order
- itunes_subtitle (string): The item subtitle

- itunes_summary (string): The summary of the item
- link (string): The URL of item.
- published_date (string): Date item was published
- title (string): The title of item.

Bugs & Feature Requests

<https://github.com/jrigden/pyPodcastParser/issues/new>

Credits

5.1 Jason Rigden

Email: jasonrigden@gmail.com

Linkedin: <https://www.linkedin.com/in/jasonrigden>

Twitter: [mr_rigden](https://twitter.com/mr_rigden)

Development

<https://github.com/jrigden/pyPodcastParser>

Testing

License

The MIT License (MIT) Copyright (c) 2016 Jason Rigden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.1 API

These are the details:

class `pyPodcastParser.Podcast.Podcast` (*feed_content*)
Parses an xml rss feed

RSS Specs <http://cyber.law.harvard.edu/rss/rss.html>

More RSS Specs <http://www.rssboard.org/rss-specification>

iTunes Podcast Specs <http://www.apple.com/itunes/podcasts/specs.html>

The cloud element aka RSS Cloud is not supported as it has been superseded by the superior PubSubHubbub protocol

Parameters `feed_content` (*str*) – An rss string

Note: All attributes with empty or nonexistent element will have a value of None

Attributes are generally strings or lists of strings, because we want to record the literal value of elements.

feed_content
str

The actual xml of the feed

soup

bs4.BeautifulSoup

A soup of the xml with items and image removed

image_soup

bs4.BeautifulSoup

soup of image

full_soup

bs4.BeautifulSoup

A soup of the xml with items

categories

list

List for strings representing the feed categories

copyright

str

The feed's copyright

creative_commons

str

The feed's creative commons license

items

item

Item objects

description

str

The feed's description

generator

str

The feed's generator

image_title

str

Feed image title

image_url

str

Feed image url

image_link

str

Feed image link to homepage

image_width

str

Feed image width

image_height*str*

Feed image height

itunes_author_name*str*

The podcast's author name for iTunes

itunes_block*bool*

Does the podcast block itunes

itunes_categories*list*

List of strings of itunes categories

itunes_complete*str*

Is this podcast done and complete

itunes_explicit*str*

Is this item explicit. Should only be "yes" and "clean."

itunes_image*str*

URL to itunes image

itunes_keywords*list*

List of strings of itunes keywords

itunes_new_feed_url*str*

The new url of this podcast

language*str*

Language of feed

last_build_date*str*

Last build date of this feed

link*str*

URL to homepage

managing_editor*str*

managing editor of feed

published_date

str

Date feed was published

pubsubhubbub

str

The URL of the pubsubhubbub service for this feed

owner_name

str

Name of feed owner

owner_email

str

Email of feed owner

subtitle

str

The feed subtitle

title

str

The feed title

ttl

str

The time to live or number of minutes to cache feed

web_master

str

The feed's webmaster

is_valid_rss

bool

Is this a valid RSS Feed

is_valid_podcast

bool

Is this a valid Podcast

date_time

datetime

When published

count_items ()

Counts Items in full_soup and soup. For debugging

set_categories ()

Parses and set feed categories

set_copyright ()

Parses copyright and set value

set_creative_commons ()

Parses creative commons for item and sets value

set_description()
Parses description and sets value

set_extended_elements()
Parses and sets non required elements

set_full_soup()
Sets soup and keeps items

set_generator()
Parses feed generator and sets value

set_image()
Parses image element and set values

set_is_valid_rss()
Check to if this is actually a valid RSS feed

set_itune_image()
Parses itunes images and set url as value

set_itunes()
Sets elements related to itunes

set_itunes_author_name()
Parses author name from itunes tags and sets value

set_itunes_block()
Check and see if podcast is blocked from iTunes and sets value

set_itunes_categories()
Parses and set itunes categories

set_itunes_complete()
Parses complete from itunes tags and sets value

set_itunes_explicit()
Parses explicit from itunes tags and sets value

set_itunes_keywords()
Parses itunes keywords and set value

set_itunes_new_feed_url()
Parses new feed url from itunes tags and sets value

set_language()
Parses feed language and set value

set_last_build_date()
Parses last build date and set value

set_link()
Parses link to homepage and set value

set_managing_editor()
Parses managing editor and set value

set_optional_elements()
Sets elements considered option by RSS spec

set_owner()
Parses owner name and email then sets value

set_published_date()
Parses published date and set value

set_pubsubhubbub()
Parses pubsubhubbub and email then sets value

set_required_elements()
Sets elements required by RSS spec

set_soup()
Sets soup and strips items

set_subtitle()
Parses subtitle and sets value

set_summary()
Parses summary and set value

set_title()
Parses title and set value

set_ttl()
Parses summary and set value

set_web_master()
Parses the feed's webmaster and sets value

class pyPodcastParser.Item.**Item**(*soup*)
Parses an xml rss feed

RSS Specs <http://cyber.law.harvard.edu/rss/rss.html> iTunes Podcast Specs
<http://www.apple.com/itunes/podcasts/specs.html>

Parameters **soup** (*bs4.BeautifulSoup*) – BeautifulSoup object representing a rss item

Note: All attributes with empty or nonexistent element will have a value of None

author
str
The author of the item

comments
str
URL of comments

creative_commons
str
creative commons license for this item

description
str
Description of the item.

enclosure_url
str
URL of enclosure

enclosure_type*str*

File MIME type

enclosure_length*int*

File size in bytes

guid*str*

globally unique identifier

itunes_author_name*str*

Author name given to iTunes

itunes_block*bool*

It this Item blocked from itunes

itunes_closed_captioned

(str): It is this item have closed captions

itunes_duration*str*

Duration of enclosure

itunes_explicit*str*

Is this item explicit. Should only be yes or clean.

itunes_image*str*

URL of item cover art

itunes_order*str*

Override published_date order

itunes_subtitle*str*

The item subtitle

itunes_summary*str*

The summary of the item

link*str*

The URL of item.

published_date*str*

Date item was published

title
str
The title of item.

date_time
datetime
When published

set_author()
Parses author and set value.

set_categories()
Parses and set categories

set_comments()
Parses comments and set value.

set_creative_commons()
Parses creative commons for item and sets value

set_description()
Parses description and set value.

set_enclosure()
Parses enclosure_url, enclosure_type then set values.

set_guid()
Parses guid and set value

set_itune_image()
Parses itunes item images and set url as value

set_itunes_author_name()
Parses author name from itunes tags and sets value

set_itunes_block()
Check and see if item is blocked from iTunes and sets value

set_itunes_closed_captioned()
Parses isClosedCaptioned from itunes tags and sets value

set_itunes_duration()
Parses duration from itunes tags and sets value

set_itunes_element()
Set each of the itunes elements.

set_itunes_explicit()
Parses explicit from itunes item tags and sets value

set_itunes_order()
Parses episode order and set url as value

set_itunes_subtitle()
Parses subtitle from itunes tags and sets value

set_itunes_summary()
Parses summary from itunes tags and sets value

set_link()
Parses link and set value.

set_published_date()
Parses published date and set value.

set_rss_element()
Set each of the basic rss elements.

set_title()
Parses title and set value.

A

author (Item attribute), 22

C

categories (Podcast attribute), 18

comments (Item attribute), 22

copyright (Podcast attribute), 18

count_items() (pyPodcastParser.Podcast.Podcast
method), 20

creative_commons (Item attribute), 22

creative_commons (Podcast attribute), 18

D

date_time (Item attribute), 24

date_time (Podcast attribute), 20

description (Item attribute), 22

description (Podcast attribute), 18

E

enclosure_length (Item attribute), 23

enclosure_type (Item attribute), 22

enclosure_url (Item attribute), 22

F

feed_content (Podcast attribute), 17

full_soup (Podcast attribute), 18

G

generator (Podcast attribute), 18

guid (Item attribute), 23

I

image_height (Podcast attribute), 18

image_link (Podcast attribute), 18

image_soup (Podcast attribute), 18

image_title (Podcast attribute), 18

image_url (Podcast attribute), 18

image_width (Podcast attribute), 18

is_valid_podcast (Podcast attribute), 20

is_valid_rss (Podcast attribute), 20

Item (class in pyPodcastParser.Item), 22

items (Podcast attribute), 18

itunes_image (Item attribute), 23

itunes_image (Podcast attribute), 19

itunes_author_name (Item attribute), 23

itunes_author_name (Podcast attribute), 19

itunes_block (Item attribute), 23

itunes_block (Podcast attribute), 19

itunes_categories (Podcast attribute), 19

itunes_closed_captioned (Item attribute), 23

itunes_complete (Podcast attribute), 19

itunes_duration (Item attribute), 23

itunes_explicit (Item attribute), 23

itunes_explicit (Podcast attribute), 19

itunes_keywords (Podcast attribute), 19

itunes_new_feed_url (Podcast attribute), 19

itunes_order (Item attribute), 23

itunes_subtitle (Item attribute), 23

itunes_summary (Item attribute), 23

L

language (Podcast attribute), 19

last_build_date (Podcast attribute), 19

link (Item attribute), 23

link (Podcast attribute), 19

M

managing_editor (Podcast attribute), 19

O

owner_email (Podcast attribute), 20

owner_name (Podcast attribute), 20

P

Podcast (class in pyPodcastParser.Podcast), 17

published_date (Item attribute), 23

published_date (Podcast attribute), 19

pubsubhubbub (Podcast attribute), 20

S

set_author() (pyPodcastParser.Item.Item method), 24

[set_categories\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_categories\(\) \(pyPodcastParser.Podcast.Podcast method\), 20](#)
[set_comments\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_copyright\(\) \(pyPodcastParser.Podcast.Podcast method\), 20](#)
[set_creative_commons\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_creative_commons\(\) \(pyPodcastParser.Podcast.Podcast method\), 20](#)
[set_description\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_description\(\) \(pyPodcastParser.Podcast.Podcast method\), 20](#)
[set_enclosure\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_extended_elements\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_full_soup\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_generator\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_guid\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_image\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_is_valid_rss\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itune_image\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itune_image\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_author_name\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_author_name\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_block\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_block\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_categories\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_closed_captioned\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_complete\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_duration\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_element\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_explicit\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_explicit\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_keywords\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_new_feed_url\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_itunes_order\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_subtitle\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_itunes_summary\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_language\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_last_build_date\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_link\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_link\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_managing_editor\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_optional_elements\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_owner\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_published_date\(\) \(pyPodcastParser.Item.Item method\), 24](#)
[set_published_date\(\) \(pyPodcastParser.Podcast.Podcast method\), 21](#)
[set_pubsubhubbub\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_required_elements\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_rss_element\(\) \(pyPodcastParser.Item.Item method\), 25](#)
[set_soup\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_subtitle\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_summary\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_title\(\) \(pyPodcastParser.Item.Item method\), 25](#)
[set_title\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_ttl\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[set_web_master\(\) \(pyPodcastParser.Podcast.Podcast method\), 22](#)
[soup \(Podcast attribute\), 18](#)
[subtitle \(Podcast attribute\), 20](#)

T

[title \(Item attribute\), 23](#)
[title \(Podcast attribute\), 20](#)
[ttl \(Podcast attribute\), 20](#)

W

[web_master \(Podcast attribute\), 20](#)