

---

# **GenRSS Documentation**

***Release 1.0.0***

**Dmitriy Pleshevskiy**

**Nov 08, 2020**



---

## Contents

---

<b>1</b>	<b>User's Guide</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Quickstart . . . . .	4
<b>2</b>	<b>API Reference</b>	<b>5</b>
2.1	API . . . . .	5
<b>3</b>	<b>Additional notes</b>	<b>9</b>
3.1	Changelog . . . . .	9
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



This is generator of rss feed for your channel. Easy for config and usage.

Get started with [Installation](#) and then get an overview with the [Quickstart](#). The rest of the docs describe each component of GenRSS in detail, with a full reference in the [API](#) section.



## 1.1 Installation

### 1.1.1 Python Version

We recommend using the latest version of Python 3. GenRSS supports Python 3.6 and newer.

### 1.1.2 Dependencies

genrss using lxml library that required:

- libxml2 version 2.9.2 or later.
- libxslt version 1.1.27 or later.

To install the required development packages of these dependencies on Linux systems, use your distribution specific installation tool, e.g. apt-get on Debian/Ubuntu:

```
sudo apt-get install libxml2-dev libxslt-dev
```

### 1.1.3 Install GenRSS

Use the following command to install GenRSS:

```
pip install -U genrss
```

GenRSS is now installed. Check out [Quickstart](#)

### 1.1.4 Install from source

If you want to work with the latest GenRSS code before it's released, install or update the code from the master branch:

```
pip install -U https://github.com/icetemple/genrss/archive/master.zip
```

## 1.2 Quickstart

Do you want to get started? This page gives a good introduction to GetRSS. It assumes you already have GenRSS installed. If you do not, head over to the *Installation* section.

### 1.2.1 A Simple Example

```
from datetime import datetime
from genrss import GenRSS

feed = GenRSS(title='SmartFridge',
              site_url='https://smartfridge.me',
              feed_url='https://smartfridge.me/feed/rss.xml')

feed.item(title='black buns for burgers',
          description='For the first time black burgers appeared ' \
                    'in Japan. Unusual dark color buns complemented ' \
                    'with black cheese and sauce. Over time, the dish ' \
                    'has conquered the whole world. The main secret ' \
                    'ingredient in popular buns is the most common... ' \
                    'charcoal!',
          url='https://smartfridge.me/recipe/316b28-chernye-bulochnki-dlya-burgerov/',
          author='@smartfridge',
          categories=['baking'],
          pub_date=datetime.utcnow())

xml = feed.xml()
```

This code initializes small feed about recipe site with one item and returns xml as string

For more information about GenRSS you can have a look [API](#) section.



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

## 2.1 API

### 2.1.1 GenRSS

**class** `genrss.GenRSS` (*title: str, site\_url: str, feed\_url: str, \*\*kwargs*)

Generates RSS feed of channel.

#### Parameters

- **title** – Title of your site or feed
- **site\_url** – Absolute url to the site that the feed is for
- **feed\_url** – Absolute url to the rss feed
- **description** – A short description of feed
- **image\_url** – Image absolute url for channel
- **image** – Image element for channel (it replaces image\_url)
- **author** – Author of channel
- **pub\_date** – Datetime in utc when last item was published
- **copyright** – Copyright information for this feed
- **language** – The language of the content of this feed.
- **editor** – Who manages content in this feed
- **webmaster** – Who manages feed availability and technical support
- **docs\_url** – Url to rss documentation
- **categories** – List of category names

- **generator** – Feed generator

**item** (*title: str, \*\*kwargs*)

Adds item to the feed.

An item can be used for recipes, blog entries, project update, log entry, etc. Your RSS feed can have any number of items.

#### Parameters

- **title** – Title of this particular item
- **description** – Content for the item. Can contain html but link and image urls must be absolute path including hostname
- **url** – Url to the item. This could be a blog entry
- **guid** – A unique string feed readers use to know if an item is new or has already been seen. If you use a guid never change it. If you don't provide a guid then your item urls must be unique
- **author** – If included it is the name of the item's creator. If not provided the item author will be the same as the feed author. This is typical except on multi-author blogs
- **categories** – If provided, each array item will be added as a category element
- **enclosure** – An enclosure object
- **pub\_date** – The date and time of when the item was created. Feed readers use this to determine the sort order. Some readers will also use it to determine if the content should be presented as unread

**to\_element** () → ElementT

Returns root element for xml.

**xml** (*pretty: bool = False*) → str

Returns the XML as a string.

Parameters **pretty** – Pretty print xml

## 2.1.2 Item

**class** `genrss.Item` (*title: str, \*\*kwargs*)

Data for item tag for rss.

#### Parameters

- **title** – Title of this particular item
- **description** – Content for the item. Can contain html but link and image urls must be absolute path including hostname
- **url** – Url to the item. This could be a blog entry
- **guid** – A unique string feed readers use to know if an item is new or has already been seen. If you use a guid never change it. If you don't provide a guid then your item urls must be unique
- **author** – If included it is the name of the item's creator. If not provided the item author will be the same as the feed author. This is typical except on multi-author blogs
- **categories** – If provided, each array item will be added as a category element
- **enclosure** – An enclosure object

- **image** – An image object
- **pub\_date** – The date and time of when the item was created. Feed readers use this to determine the sort order. Some readers will also use it to determine if the content should be presented as unread

**to\_element** () → ElementT  
Returns item element for xml.

### 2.1.3 Enclosure

**class** genrss.**Enclosure** (*url: str, size=None, type=None*)  
Data for enclosure tag for rss.

#### Parameters

- **url** – Absolute url to file
- **size** – File size
- **type** – File mime type

**classmethod from\_dict** (*data: Dict[str, Union[str, int]]*)  
Makes enclosure data from dict.

**to\_element** () → ElementT  
Returns item element for xml.



legal information and changelog are here for the interested.

## 3.1 Changelog

### 3.1.1 Version 1.0.6

Released 2020-11-08

- Fix channel image

### 3.1.2 Version 1.0.5

Released 2020-11-06

- Update lxml version

### 3.1.3 Version 1.0.3

Released 2019-07-30

- Added Item and Enclosure objects.
- Fixed bug with encodings.

### 3.1.4 Version 1.0.0

Released 2019-07-23

- First public preview release.



**g**

genrss, [5](#)





## E

`Enclosure` (*class in genrss*), 7

## F

`from_dict()` (*genrss.Enclosure class method*), 7

## G

`GenRSS` (*class in genrss*), 5

`genrss` (*module*), 5

## I

`Item` (*class in genrss*), 6

`item()` (*genrss.GenRSS method*), 6

## T

`to_element()` (*genrss.Enclosure method*), 7

`to_element()` (*genrss.GenRSS method*), 6

`to_element()` (*genrss.Item method*), 7

## X

`xml()` (*genrss.GenRSS method*), 6