

---

# **comics Documentation**

*Release 2.4.3*

**Stein Magnus Jodal**

**Jan 22, 2019**



<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Installation	3
1.1.1	Get <i>comics</i>	3
1.1.2	Software requirements	3
1.1.3	Run <i>comics</i>	5
1.2	Deployment	6
1.2.1	Database	6
1.2.2	Example Apache vhost	7
1.2.3	Available settings	7
1.2.4	Example <code>settings/local.py</code>	8
1.2.5	Collecting static files	9
1.2.6	Example cronjob	9
1.3	Creating new crawlers	10
1.3.1	A crawler example	10
1.3.2	The <code>ComicData</code> class fields	10
1.3.3	The <code>Crawler</code> class fields	11
1.3.4	The <code>Crawler.crawl()</code> method	12
1.3.5	<code>LxmlParser</code> – Parsing web pages and HTML	13
1.3.6	<code>FeedParser</code> – Parsing feeds	15
1.3.7	Testing your new crawler	16
1.3.8	Submitting your new crawler for inclusion in <i>comics</i>	16
1.4	Web service	17
1.4.1	Authentication	17
1.4.2	Response format	17
1.4.3	Pagination	18
1.4.4	Resources	19
1.5	Development	28
1.5.1	How to contribute	28
1.5.2	Data model	29
1.5.3	Running tests	31
1.6	Authors	31
1.7	Changes	31
1.7.1	v2.4.3 (2017-04-01)	31
1.7.2	v2.4.2 (2016-01-19)	32
1.7.3	v2.4.1 (2015-09-04)	32
1.7.4	v2.4.0 (2015-01-03)	33

1.7.5	v2.3.4 (2015-01-03)	33
1.7.6	v2.3.3 (2014-10-06)	34
1.7.7	v2.3.2 (2014-09-11)	34
1.7.8	v2.3.1 (2014-06-11)	35
1.7.9	v2.3.0 (2014-04-07)	36
1.7.10	v2.2.3 (2014-03-31)	36
1.7.11	v2.2.2 (2013-12-21)	36
1.7.12	v2.2.1 (2013-11-08)	37
1.7.13	v2.2.0 (2013-07-07)	38
1.7.14	v2.1.1 (2013-02-26)	40
1.7.15	v2.1.0 (2012-10-15)	40
1.7.16	v2.0.1 (2012-10-06)	41
1.7.17	v2.0.0 (2012-06-11)	41
1.7.18	v1.1.6 (2012-06-10)	43
1.7.19	v1.1.5 (2012-05-09)	45
1.7.20	v1.1.4 (2012-04-07)	45
1.7.21	v1.1.3 (2012-01-29)	46
1.7.22	v1.1.2 (2011-09-18)	48
1.7.23	v1.1.1 (2011-08-22)	48
1.7.24	v1.1.0 (2011-08-15)	48
1.7.25	v1.0.8 (2011-08-10)	50
1.7.26	v1.0.7 (2011-07-13)	50
1.7.27	v1.0.6 (2011-02-19)	51
1.7.28	v1.0.5 (2010-12-29)	52
1.7.29	v1.0.4 (2010-10-23)	53
1.7.30	v1.0.3 (2010-07-26)	53
1.7.31	v1.0.2 (2010-04-11)	54
1.7.32	v1.0.1 (2010-02-23)	55
1.7.33	v1.0.0 (2010-01-27)	56
1.7.34	v1.0.0.rc2 (2010-01-19)	56
1.7.35	v1.0.0.rc1 (2010-01-17)	56
<b>2</b>	<b>Indices and tables</b>	<b>57</b>
	<b>HTTP Routing Table</b>	<b>59</b>
	<b>Python Module Index</b>	<b>61</b>

*comics* is a web comics aggregator. Out of the box it can crawl and archive about two hundred comics every day. The comics are made available through an easy to use web interface where users can build personalized collections of their favorite comics, and then read them on the site or export them using feeds.

Adding a new comic to your installation requires only the addition of a single Python file with some metadata and a few lines of code. To make crawler development easy, *comics* comes with both documentation and powerful APIs for crawling web sites and feeds.

*comics* runs on Python and Django. It is licensed under the GNU Affero General Public License version 3.

- [Source code](#)
- [Documentation](#)
- [Issues](#)
- [Contributors](#)



## 1.1 Installation

First of all, *comics* is just a Django application. Thus, if there are details not outlined in *comics*' own docs, you'll probably find the answer in Django's docs. For example, database settings are mentioned on this page, but no details are given, as we're just talking about Django's database settings. Django got better docs for their database settings than we could ever write, so please refer to Django's docs.

### 1.1.1 Get *comics*

You can get hold of *comics* in two ways:

- Download the latest release from <http://github.com/jodal/comics/tags> and unpack it.
- Clone the Git repository. You can do so by running:

```
git clone https://github.com/jodal/comics.git
cd comics/
```

You'll then find the current stable/maintenance version in the `master` branch:

```
git checkout master
```

And the current development version in the `develop` branch:

```
git checkout develop
```

### 1.1.2 Software requirements

The dependencies can be installed using [pip](#).

You can either install the dependencies globally on your computer:

```
cd comics/  
sudo pip install -r requirements.txt
```

Or, in an isolated environment using `virtualenv`:

```
cd comics/  
virtualenv ../comics-virtualenv/  
source ../comics-virtualenv/bin/activate  
pip install -r requirements.txt
```

If you make use of a `virtualenv` for a real deployment, you'll also need to make sure that the WSGI file and the cronjob activate the `virtualenv`.

### Minimum dependencies

The absolute minimum requirements for getting `comics` up and running are documented in the file `requirements.txt`:

```
Django >= 1.7, < 1.8  
  
# For timezone travel  
pytz  
  
# For image handling  
Pillow >= 1.7, < 2.8  
  
# For crawling comics and the REST API  
lxml >= 3, < 4  
  
# For crawling comics  
feedparser >= 4.0, < 6  
cssselect >= 0.8, < 0.10  
  
# For static resource compression  
django_compressor >= 1.1, < 1.5  
cssmin >= 0.1, < 0.3  
jsmin >= 2.0, < 2.1  
  
# For user registration with email verification  
django-registration >= 0.8, < 0.9  
  
# For user invitation  
-e ./django-invitation  
  
# For Twitter Bootstrap compatible Django forms  
django-bootstrap-form >= 3.1, < 3.2  
  
# For REST API  
django-tastypie >= 0.12, < 0.13  
defusedxml >= 0.4, < 0.5
```

### Optional dependencies for real deployments

For a real deployment, you should consider using another database than SQLite, which is the default. In that case, you also need to install Python libraries for connecting to your database of choice, e.g. `psycopg2` if you are using



PostgreSQL.

### Optional dependencies for development

There are also some additional requirements only needed for development, which are listed in the file `requirements-dev.txt`:

```
# Automate tasks
invoke

# Mocking in unit tests
mock

# Debugging of database queries, etc.
django-debug-toolbar

# Generation of the model graph used in the docs
django-extensions
```

### 1.1.3 Run *comics*

To get *comics* to a state useful for testing of new crawlers and personal usage, the following steps are all that is needed.

#### Create database

A file-based SQLite database will be used, unless you have created a file `comics/comics/settings/local.py` where you have configured another database, like PostgreSQL.

To create the database and database schema, open a terminal, go to top level directory in your checkout of the *comics* repo, where you'll find the file `manage.py`, and run:

```
python manage.py migrate --noinput
```

`--noinput` stops `syncdb` from asking you to create a superuser, as this will fail at this point. Instead, when `syncdb` has finished, create a superuser by running:

```
python manage.py createsuperuser
```

#### Seed database

Then we need to seed the database with information on what comics to crawl. E.g. to add the *XKCD* comic from `comics/comics/comics/xkcd.py`, run:

```
python manage.py comics_addcomics -c xkcd
```

Optionally, you can add all available active comics to the database:

```
python manage.py comics_addcomics -c all
```

### Get some comic releases

Next, we need to get hold of some comic releases, so we will crawl the web for them. This will get today's releases for all added comics:

```
python manage.py comics_getreleases
```

To get the release for a specific added comics, you can filter with `--comic` or `-c`:

```
python manage.py comics_getreleases -c xkcd
```

To get releases for a range of days, you can specify a date range with `--from` or `-f` and `--to` or `-t`. Both defaults to today, so you can leave the end of the range out:

```
python manage.py comics_getreleases -f 2011-11-11
```

### Development web server

Finally, to be able to browse the comic releases we have aggregated, start the Django development web server by running:

```
python manage.py runserver --settings=comics.settings.dev
```

If you now point your web browser at <http://localhost:8000/> you will be able to browse all available comics. If you provided a username and password at the `syncdb` step, you can log in at <http://localhost:8000/admin/> to do simple administration tasks, like removing comics or releases.

### More options

All of the `manage.py` commands got more options available. Add the `--help` argument to any of the commands to get a full listing of the available options.

## 1.2 Deployment

The following example documents *one way* to deploy *comics*. As *comics* is a standard Django project with an additional batch job for , it may be deployed in just about any way a Django project may be deployed. Please refer to [Django's deployment documentation](#) for further details.

In the following examples we assume that we are deploying *comics* at <http://comics.example.com/>, using Apache, `mod_wsgi`, and PostgreSQL. The Django application and batch job is both running as the user `comics-user`. The static media files, like comic images, are served from <http://comics.example.com/media/>, but may also be served from a different host.

### 1.2.1 Database

*comics* should theoretically work with any database supported by Django. Though, development is mostly done on SQLite 3 and PostgreSQL 8.x. For production use, PostgreSQL is the recommended choice.

---

**Note:** If you are going to use SQLite in a deployment with Apache and so on, you need to ensure that the user the web server will be running as has write access to the *directory* the SQLite database file is located in.

---

## Additional database indexes

Out of the box, *comics* will create a few extra database indexes that will make it a lot more performant. In addition, creating the following indexes will improve performance a bit more:

```
CREATE INDEX comics_release_comic_id_pub_date
ON comics_release (comic_id, pub_date);
```

## 1.2.2 Example Apache vhost

This example requires your Apache to have the `mod_wsgi` module. For efficient static media serving and caching, you should probably enable `mod_deflate` and `mod_expires` for `/media` and `/static`.

```
<VirtualHost *:80>
  ServerName comics.example.com
  ErrorLog /var/log/apache2/comics.example.com-error.log
  CustomLog /var/log/apache2/comics.example.com-access.log combined

  # Not used, but Apache will complain if the dir does not exist
  DocumentRoot /var/www/comics.example.com

  # Static media hosting
  Alias /media/ /path/to/comics/media/
  Alias /static/ /path/to/comics/static/

  # mod_wsgi setup
  WSGIDaemonProcess comics user=comics-user group=comics-user threads=50 maximum-
  ↪requests=10000
  WSGIProcessGroup comics
  WSGIScriptAlias / /path/to/comics/comics/wsgi/__init__.py
  <Directory /path/to/comics/comics/wsgi>
    Require all granted
  </Directory>
</VirtualHost>
```

For details, please refer to the documentation of the `Apache` and `mod_wsgi` projects.

## 1.2.3 Available settings

The Django docs got a [list of all available settings](#). Some of them are repeated here if they are especially relevant for *comics*. In addition, *comics* adds some settings of its own, which are all listed here as well.

```
comics.settings.base.ACCOUNT_ACTIVATION_DAYS = 7
    Number of days an the account activation link will work
```

```
comics.settings.base.ACCOUNT_INVITATION_DAYS = 7
    Number of days an invitation will be valid
```

```
comics.settings.base.COMICS_BROWSER_REFRESH_INTERVAL = 60
    Number of seconds browsers at the latest view of “My comics” should wait before they check for new releases again
```

```
comics.settings.base.COMICS_GOOGLE_ANALYTICS_CODE = None
    Google Analytics tracking code. Tracking code will be included on all pages if this is set.
```

```
comics.settings.base.COMICS_IMAGE_BLACKLIST = ('e3b0c44298fc1c149afb4c8996fb92427ae41e464
    SHA256 of blacklisted images
```

`comics.settings.base.COMICS_LOG_FILENAME = '/home/docs/checkouts/readthedocs.org/user_builds/comics/logfile'`  
Comics log file path on disk

`comics.settings.base.COMICS_MAX_DAYS_IN_FEED = 30`  
Maximum number of days to show in a feed

`comics.settings.base.COMICS_MAX_RELEASES_PER_PAGE = 50`  
Maximum number of releases to show on one page

`comics.settings.base.COMICS_NUM_DAYS_COMIC_IS_NEW = 7`  
Number of days a new comic on the site is labeled as new

`comics.settings.base.COMICS_SITE_TITLE = 'example.com'`  
Name of the site. Used in page header, page title, feed titles, etc.

`comics.settings.base.DATABASES = {'default': {'ENGINE': 'django.db.backends.sqlite3', 'NAME': 'db.sqlite3'}}`  
Database settings. You will want to change this for production. See the Django docs for details.

`comics.settings.base.INVITATIONS_PER_USER = 10`  
Number of invitations each existing user can send

`comics.settings.base.INVITE_MODE = False`  
Turn invitations off by default, leaving the site open for user registrations

`comics.settings.base.MEDIA_ROOT = '/home/docs/checkouts/readthedocs.org/user_builds/comics/media'`  
Path on disk to where downloaded media will be stored and served from

`comics.settings.base.MEDIA_URL = '/media/'`  
URL to where downloaded media will be stored and served from

`comics.settings.base.SESSIION_COOKIE_AGE = 31536000`  
Time the user session cookies will be valid. 1 year by default.

`comics.settings.base.STATIC_ROOT = '/home/docs/checkouts/readthedocs.org/user_builds/comics/static'`  
Path on disk to where static files will be served from

`comics.settings.base.STATIC_URL = '/static/'`  
URL to where static files will be served from

`comics.settings.base.TIME_ZONE = 'UTC'`  
Default time zone to use when displaying datetimes to users

### 1.2.4 Example settings/local.py

To change settings, you should not change the settings files shipped with *comics*, but instead override the settings in your own `comics/comics/settings/local.py`. Even if you do not want to override any default settings, you must add a `local.py` which at least sets `SECRET_KEY` and most probably your database settings. A full `local.py` may look like this:

```
# Local settings -- do NOT commit to a VCS

# Make this unique, and don't share it with anybody.
SECRET_KEY = 'djdjdk5k4$(DA=!SDAD!)12312415151368edkfgjngnw3m!$!Dfafa'

# You can override any settings here, like database settings.

ADMINS = (
    ('Comics Webmaster', 'comics@example.com'),
)
MANAGERS = ADMINS
```

(continues on next page)

(continued from previous page)

```

# Database settings
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'comics',
        'USER': 'comics',
        'PASSWORD': 'topsecret',
        'HOST': 'localhost',
        'PORT': '',
    }
}

# Internal IP addresses
INTERNAL_IPS = ('127.0.0.1',)

# Media
MEDIA_ROOT = '/var/www/comics.example.com/media/'
MEDIA_URL = 'http://comics.example.com/media/'
STATIC_ROOT = '/var/www/comics.example.com/static/'
STATIC_URL = 'http://comics.example.com/static/'

# Caching
CACHE_BACKEND = 'memcached://127.0.0.1:11211/'
CACHE_MIDDLEWARE_KEY_PREFIX = 'comics'

```

Of course, you should change most, if not all, of these settings for your own installation. If your are not running a *memcached* server, remove the part on caching from your `local.py`.

## 1.2.5 Collecting static files

When you're not running in development mode, you'll need to collect the static files from all apps into the `STATIC_ROOT`. To do this, run:

```
python manage.py collectstatic
```

You have to rerun this command every time you deploy changes to graphics, CSS and JavaScript. For more details, see the Django documentation on [staticfiles](#).

## 1.2.6 Example cronjob

To get new comics, you should run `comics_getreleases` regularly. In addition, you should run `cleanupinvitation` once in a while to remove expired user invitations and `cleanupregistration` to delete expired users. One way is to use `cron` e.g. by placing the following in `/etc/cron.d/comics`:

```

MAILTO=comics@example.com
PYTHONPATH=/path/to/comics
1 * * * * comics-user python /path/to/comics/manage.py comics_getreleases -v0
1 3 * * * comics-user python /path/to/comics/manage.py cleanupinvitation -v0
2 3 * * * comics-user python /path/to/comics/manage.py cleanupregistration -v0

```

If you have installed *comics*' dependencies in a `virtualenv` instead of globally, the cronjob must also activate the `virtualenv`. This can be done by using the `python` interpreter from the `virtualenv`:

```
MAILTO=comics@example.com
PYTHONPATH=/path/to/comics
1 * * * * comics-user /path/to/comics-virtualenv/bin/python /path/to/comics/manage.py_
↳comics_getreleases -v0
1 3 * * * comics-user /path/to/comics-virtualenv/bin/python /path/to/comics/manage.py_
↳cleanupinvitation -v0
2 3 * * * comics-user /path/to/comics-virtualenv/bin/python /path/to/comics/manage.py_
↳cleanupregistration -v0
```

By setting MAILTO any exceptions raised by the comic crawlers will be sent by mail to the given mail address. 1 \* \* \* \* specifies that the command should be run 1 minute past every hour.

## 1.3 Creating new crawlers

For each comic *comics* is aggregating, we need to create a crawler. At the time of writing, about 100 crawlers are available in the `comics/comics/comics/` directory. They serve as a great source for learning how to write new crawlers for *comics*.

### 1.3.1 A crawler example

The crawlers are split in two separate pieces. The *ComicData* part contains meta data about the comic used for display at the web site. The *Crawler* part contains properties needed for crawling and the crawler implementation itself.

```
from comics.aggregator.crawler import CrawlerBase, CrawlerImage
from comics.core.comic_data import ComicDataBase

class ComicData(ComicDataBase):
    name = 'xkcd'
    language = 'en'
    url = 'http://www.xkcd.com/'
    start_date = '2005-05-29'
    rights = 'Randall Munroe, CC BY-NC 2.5'

class Crawler(CrawlerBase):
    history_capable_days = 10
    schedule = 'Mo,We,Fr'
    time_zone = 'US/Eastern'

    def crawl(self, pub_date):
        feed = self.parse_feed('http://www.xkcd.com/rss.xml')
        for entry in feed.for_date(pub_date):
            url = entry.summary.src('img[src*="/comics/"]')
            title = entry.title
            text = entry.summary.alt('img[src*="/comics/"]')
            return CrawlerImage(url, title, text)
```

### 1.3.2 The ComicData class fields

```
class ComicData
```

**name**  
*Required.* A string with the name of the comic.

**url**  
*Required.* A string with the URL of the comic's web page.

**active**  
*Optional.* Whether or not this comic is still being crawled. Defaults to `True`.

**start\_date**  
*Optional.* The first date the comic was published at.

**end\_date**  
*Optional.* The last date the comic was published at if it is discontinued.

**rights**  
*Optional.* Name of the author and the comic's license if available.

### 1.3.3 The Crawler class fields

**class Crawler**

**history\_capable\_date**  
*Optional.* Date of oldest release available for crawling.  
 Provide this *or* `Crawler.history_capable_days`. If both are present, this one will have precedence.  
 Example: `'2008-03-08'`.

**history\_capable\_days**  
*Optional.* Number of days a release is available for crawling.  
 Provide this *or* `Crawler.history_capable_date`.  
 Example: `32`.

**schedule**  
*Optional.* On what weekdays the comic is published.  
 Example: `'Mo, We, Fr'` or `'Mo, Tu, We, Th, Fr, Sa, Su'`.

**time\_zone**  
*Optional.* In approximately what time zone the comic is published.  
 Example: `Europe/Oslo` or `US/Eastern`.  
 See [the IANA timezone database](#) for a list of possible values.

**multiple\_releases\_per\_day**  
*Optional.* Default: `False`. Whether to allow multiple releases per day.  
 Example: `True` or `False`.

**has\_rerun\_releases**  
*Optional.* Default: `False`. Whether the comic reruns old images as new releases.  
 Example: `True`` or `False``.

**headers**  
*Optional.* Default: `None`. Any HTTP headers to send with any URI request for values.

Useful if you're pulling comics from a site that checks either the `Referer` or `User-Agent`. If you can view the comic using your browser but not when using your loader for identical URLs, try setting the `Referer` to be `http://www.example.com/` or set the `User-Agent` to be `Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)`.

Example: `{'Referer': 'http://www.example.com/', 'Host': 'http://www.example.com/'}`

### 1.3.4 The `Crawler.crawl()` method

The `Crawler.crawl()` is where the real work is going on. To start with an example, let's look at *XKCD*'s `Crawler.crawl()` method:

```
def crawl(self, pub_date):
    feed = self.parse_feed('http://www.xkcd.com/rss.xml')
    for entry in feed.for_date(pub_date):
        url = entry.summary.src('img[src*="/comics/"]')
        title = entry.title
        text = entry.summary.alt('img[src*="/comics/"]')
        return CrawlerImage(url, title, text)
```

#### Arguments and return values

The `Crawler.crawl()` method takes a single argument, `pub_date`, which is a `datetime.date` object for the date the crawler is currently crawling. The goal of the method is to return a `CrawlerImage` object containing at least the URL of the image for `pub_date` and optionally a title and text accompanying the image. `CrawlerImage`'s signature is:

```
CrawlerImage(url, title=None, text=None)
```

This means that you must always supply an URL, and that you can supply a text without a title. The following are all valid ways to create a `CrawlerImage`:

```
CrawlerImage(url)
CrawlerImage(url, title)
CrawlerImage(url, title, text)
CrawlerImage(url, text=text)
```

For some crawlers, this is all you need. If the image URL is predictable and based upon the `pub_date` in some way, just create the URL with the help of Python's [strftime documentation](#), and return it wrapped in a `CrawlerImage`:

```
def crawl(self, pub_date):
    url = 'http://www.example.com/comics/%s.png' % (
        pub_date.strftime('%Y-%m-%d'),)
    return CrawlerImage(url)
```

Though, for most crawlers, some interaction with RSS or Atom feeds or web pages are needed. For this a *web parser* and a *feed parser* are provided.

#### Returning multiple images for a single comic release

Some comics got releases with multiple images, and thus returning a single `CrawlerImage` will not be enough for you. For situations like these, *comics* lets you return a list of `CrawlerImage` objects from `Crawler.crawl()`. The list should be ordered in the same way as the comic is meant to be read, with the first frame as the first element



in the list. If the comic release got a `title`, add it to the first `CrawlerImage` object, and let the `title` field stay empty on the rest of the list elements. The same applies for the `text` field, unless each image actually got a different `title` or `text` string.

The following is an example of a `Crawler.crawl()` method which returns multiple images. It adds a `title` to the first list element, and different `text` to all of the elements.

```
def crawl(self, pub_date):
    feed = self.parse_feed('http://feeds.feedburner.com/Pidjin')
    for entry in feed.for_date(pub_date):
        result = []
        for i in range(1, 10):
            url = entry.content0.src('img[src$="000%d.jpg"]' % i)
            text = entry.content0.title('img[src$="000%d.jpg"]' % i)
            if url and text:
                result.append(CrawlerImage(url, text=text))
        if result:
            result[0].title = entry.title
    return result
```

### 1.3.5 LxmlParser – Parsing web pages and HTML

The web parser, internally known as `LxmlParser`, uses CSS selectors to extract content from HTML. For a primer on CSS selectors, see [Matching HTML elements using CSS selectors](#).

The web parser is accessed through the `Crawler.parse_page()` method:

```
def crawl(self, pub_date):
    page_url = 'http://ars.userfriendly.org/cartoons/?id=%s' % (
        pub_date.strftime('%Y%m%d'),)
    page = self.parse_page(page_url)
    url = page.src('img[alt^="Strip for"]')
    return CrawlerImage(url)
```

This is a common pattern for crawlers. Another common patterns is to use a feed to find the web page URL for the given date, then parse that web page to find the image URL.

#### LxmlParser API

The available methods only require a CSS selector, `selector`, to match tags. In the event that the selector doesn't match any elements, default will be returned.

If the `selector` matches multiple elements, one of two things will happen:

- If `allow_multiple` is `False`, a `MultipleElementsReturned` exception is raised.
- If `allow_multiple` is `True`, a list of zero or more elements is returned with all of the elements matching `selector`.

```
class comics.aggregator.lxmlparser.LxmlParser
```

```
text(selector[, default=None, allow_multiple=False])
    Returns the text contained by the element matching selector.
```

```
src(selector[, default=None, allow_multiple=False])
    Returns the src attribute of the element matching selector.
```

The web parser automatically expands relative URLs in the source, like `/comics/2008-04-13.png` to a full URL like `http://www.example.com/2008-04-13.png`, so you do not need to think about that.

- alt** (*selector*[, *default=None*, *allow\_multiple=False* ])  
Returns the `alt` attribute of the element matching *selector*.
- title** (*selector*[, *default=None*, *allow\_multiple=False* ])  
Returns the `title` attribute of the element matching *selector*.
- href** (*selector*[, *default=None*, *allow\_multiple=False* ])  
Returns the `href` attribute of the element matching *selector*.
- value** (*selector*[, *default=None*, *allow\_multiple=False* ])  
Returns the `value` attribute of the element matching *selector*.
- id** (*selector*[, *default=None*, *allow\_multiple=False* ])  
Returns the `id` attribute of the element matching *selector*.
- remove** (*selector*)  
Remove the elements matching *selector* from the parsed document.

### Matching HTML elements using CSS selectors

Both web page and feed parsing uses CSS selectors to extract the interesting strings from HTML. CSS selectors are those normally simple strings you use in CSS style sheets to select what elements of your web page the CSS declarations should be applied to.

In the following example `h1 a` is the selector. It matches all `a` elements contained in `h1` elements. The rule to be applied to the matching elements is `color: red;`.

```
h1 a { color: red; }
```

Similarly `class="foo"` and `id="bar"` in HTML may be used in CSS selectors. The following CSS example would color all `h1` headers with the class `foo` red, and all elements with the ID `bar` which is contained in `h1` elements would be colored blue.

```
h1.foo { color red; }  
h1 #bar { color: blue; }
```

In CSS3, the power of CSS selectors have been greatly increased by the addition of matching by the content of elements' attributes. To match all `img` elements with a `src` attribute *starting with* `http://www.example.com/` simply write:

```
img[src^="http://www.example.com/"]
```

Or, to match all `img` elements whose `src` attribute *ends in* `.jpg`:

```
img[src$=".jpg"]
```

Or, `img` elements whose `src` attribute *contains* `/comics/`:

```
img[src*="/comics/"]
```

Or, `img` elements whose `alt` attribute *is* `Today's comic`:

```
img[alt="Today's comic"]
```

For further details on CSS selectors in general, please refer to <http://css.maxdesign.com.au/selectutorial/>.

### 1.3.6 FeedParser – Parsing feeds

The feed parser is initialized with a feed URL passed to `Crawler.parse_feed()`, just like the web parser is initialized with a web page URL:

```
def crawl(pub_date):
    ...
    feed = self.parse_feed('http://www.xkcd.com/rss.xml')
    ...
```

#### FeedParser API

The feed object provides two methods which both returns feed elements: `FeedParser.for_date()` and `FeedParser.all()`. Typically, a crawler uses `FeedParser.for_date()` and loops over all entries it returns to find the image URL:

```
for entry in feed.for_date(pub_date):
    # parsing comes here
    return CrawlerImage(url)
```

**class** `comics.aggregator.feedparser.FeedParser`

**for\_date** (*date*)

Returns all feed elements published at *date*.

**all** ()

Returns all feed elements.

#### Feed Entry API

The *comics* feed parser is really a combination of the popular `feedparser` library and `LxmlParser`. It can do anything `feedparser` can do, and in addition you can use the `LxmlParser` methods on feed fields which contains HTML, like `Entry.summary` and `Entry.content0`.

**class** `comics.aggregator.feedparser.Entry`

**summary**

This is the most frequently used entry field which supports HTML parsing with the `LxmlParser` methods.

Example usage:

```
url = entry.summary.src('img')
title = entry.summary.alt('img')
```

**content0**

This is the same as `feedparser`'s `content[0].value` field, but with `LxmlParser` methods available. For some crawlers, this is where the interesting stuff is found.

**html** (*string*)

Wrap *string* in a `LxmlParser`.

If you need to parse HTML in any other fields than `summary` and `content0`, you can apply the `html(string)` method on the field, like it is applied on a feed entry's title field here:

```
title = entry.html(entry.title).text('h1')
```

#### tags

List of tags associated with the entry.

### 1.3.7 Testing your new crawler

When the first version of your crawler is complete, it's time to test it.

The file name is important, as it is used as the comic's slug. This means that it must be unique within the *comics* installation, and that it is used in the URLs *comics* will serve the comic at. For this example, we call the crawler file `foo.py`. The file must be placed in the `comics/comics/comics/` directory, and will be available in Python as `comics.comics.foo`.

#### Loading ComicData for your new comic

For *comics* to know about your new crawler, you need to load the comic meta data into *comics*'s database. To do so, we run the `comics_addcomics` command:

```
python manage.py comics_addcomics -c foo
```

If you do any changes to the `ComicData` class of any crawler, you must rerun `comics_addcomics` to update the database representation of the comic.

#### Running the crawler

When `comics_addcomics` has created a `comics.core.models.Comic` instance for the new crawler, you may use your new crawler to fetch the comic's release for the current date by running:

```
python manage.py comics_getreleases -c foo
```

If you want to get comics releases for more than the current day, you may specify a date range to crawl, like:

```
python manage.py comics_getreleases -c foo -f 2009-01-01 -t 2009-03-31
```

The date range will automatically be adjusted to the crawler's *history capability*. You may also get comics for a date range without a specific end. In which case, the current date will be used instead:

```
python manage.py comics_getreleases -c foo -f 2009-01-01
```

If your new crawler is not working properly, you may add `-v2` to the command to turn on full debug output:

```
python manage.py comics_getreleases -c foo -v2
```

For a full overview of `comics_getreleases` options, run:

```
python manage.py comics_getreleases --help
```

### 1.3.8 Submitting your new crawler for inclusion in *comics*

When your crawler is working properly, you may submit it for inclusion in *comics*. You should fork *comics* at [GitHub](#), commit your new crawler to your own fork, and send me a *pull request* through GitHub.

All contributions must be granted under the same license as *comics* itself.

## 1.4 Web service

*comics* comes with a web service that exposes all useful data about the current user, the user's comics subscriptions, comics, comic releases, and comic images. The web service may be used to e.g. create iOS/Android apps or alternative comics browsers, while leaving the comics crawling job to a *comics* instance.

Please make any apps using this API generic, so that they can be used with any *comics* instance as the backend. In other words, when starting your app, let the end user enter the hostname of the *comics* instance, in addition to his secret key or email/password pair.

### 1.4.1 Authentication

The web service is only available for users with an active user account on the *comics* instance. The user must authenticate himself using the same secret key as is used to access comic feeds. The key can be found in the account section of the *comics* instance.

The secret key can be provided in one of two ways:

- Using a HTTP GET parameter named `key`, i.e. as part of the URL. Example:

```
http://example.com/api/v1/users/?key=76acdcd16ae4e12becb00d09a9d9456
```

- Using the Authorization HTTP header. Example:

```
Authorization: Key 76acdcd16ae4e12becb00d09a9d9456
```

### Get secret key using email and password

If it's inconvenient for the end user to enter the secret key in your user interface—on mobile phones copy-pasting the API key from the *comics* instance's account page is time consuming at best—you may retrieve the secret key on behalf of the end user by following steps:

1. Ask the end user to provide:
  - the *comics* instance's base URL (e.g. `example.com`)
  - the email address the end user have registered on the *comics* instance
  - the password the end user have registered on the *comics* instance
2. Use the provided information to retrieve the *Users resource* from the API. Authenticate using [Basic Authentication](#) with the email address as username and the password as password. This does only work for the *Users resource*.
3. In the response from the *Users resource*, you'll find the end user's secret key. You should cache the secret key in your application and use it for all future requests to the API on behalf of this user. The end user's password should be thrown away at this point.

### 1.4.2 Response format

You can specify what response format you prefer in one of two ways:

- Using a HTTP GET parameter named `format`, i.e. as part of the URL. Examples:

```
# Returns JSON
http://example.com/api/v1/?format=json

# Returns JSONP with function name 'callback'
http://example.com/api/v1/?format=jsonp

# Returns JSONP with function name 'foo'
http://example.com/api/v1/?format=jsonp&callback=foo

# Returns JSONP with function name 'foo'
http://example.com/api/v1/?callback=foo

# Returns XML
http://example.com/api/v1/?format=xml

# Returns YAML
http://example.com/api/v1/?format=yaml

# Returns Apple binary plist
http://example.com/api/v1/?format=plist
```

- Using the Accept HTTP header. Examples:

```
# Returns JSON
Accept: application/json

# Returns JSONP with function name 'callback'
Accept: text/javascript

# Returns XML
Accept: application/xml

# Returns YAML
Accept: text/yaml

# Returns Apple binary plist
Accept: application/x-plist
```

JSON and JSONP are always supported. Other formats like XML, YAML, and Apple binary plists (bplist) may be available if the *comics* instance got the additional dependencies required by the format installed.

If you run a *comics* instance yourself, and want support for more response formats, check out [Tastypie's serialization docs](#) for details on what you need to install.

### 1.4.3 Pagination

All the resource collections support pagination. The pagination parameters that may be passed as [GET](#) arguments are:

- **limit** – max number of returned resources per response. Defaults to 20. Use 0 to remove the limit and request all objects in a single response.
- **offset** – offset into the full collection of resources. Defaults to 0.

The meta section of the collection responses include the current pagination parameters, and—if available—links to the previous and next page, and the total count of resources matching the query:

- **next** – link to the next page of the collection, if available
- **previous** – link to the previous page of the collection, if available

- **total\_count** – total number of resources in the collection, given the current filters.

## 1.4.4 Resources

### Root resource

#### GET /api/v1/

Lists all available resources, and URLs for their schemas.

### Users resource

#### GET /api/v1/users/

List of all authenticated users. Not surprisingly, it always has a single result.

#### Example request using secret key

```
GET /api/v1/users/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

#### Example request using Basic Authentication

This is the only resource that also accepts [Basic Authentication](#), using the user's email address and password. Use the secret key from the response for authenticating all future requests to the API.

```
GET /api/v1/users/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Basic YWxpY2VAZXhhbXBsZS5jb206c2VjcmV0
```

#### Example response

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  meta: {
    limit: 20,
    next: null,
    offset: 0,
    previous: null,
    total_count: 1
  },
  objects: [
    {
      date_joined: "2012-04-30T18:39:59+00:00",
      email: "alice@example.com",
      last_login: "2012-06-09T23:09:54.312109+00:00",
      resource_uri: "/api/v1/users/1/",
      secret_key: "76acdcdf16ae4e12becb00d09a9d9456"
    }
  ]
}
```

### Status Codes

- 200 OK – no error
- 401 Unauthorized – authentication/authorization failed

### Comics resource

#### GET /api/v1/comics/

Lists all available comics. Supports *Pagination*.

#### Example request

```
GET /api/v1/comics/?slug=xkcd HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcd16ae4e12becb00d09a9d9456
```

#### Example response

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  meta: {
    limit: 20,
    next: null,
    offset: 0,
    previous: null,
    total_count: 1
  },
  objects: [
    {
      active: true,
      added: "0001-01-01T00:00:00+00:00",
      end_date: null,
      id: "18",
      language: "en",
      name: "xkcd",
      resource_uri: "/api/v1/comics/18/",
      rights: "Randall Munroe, CC BY-NC 2.5",
      slug: "xkcd",
      start_date: "2005-05-29",
      url: "http://www.xkcd.com/"
    }
  ]
}
```

### Query Parameters

- **subscribed** – only include comics the authorized user is subscribed to if `true`, or unsubscribed to if `false`
- **active** – only include active comics (`true`) or inactive comics (`false`)
- **language** – only include comics with the exact language, e.g. `en`
- **name** – only include comics with matching name. Queries like `name__startswith=Dilbert` and `name__iexact=Xkcd` are supported.



- **slug** – only include comics with matching slug. Queries like `slug__contains=kc` and `slug__endswith=db` are supported.

#### Status Codes

- **200 OK** – no error
- **400 Bad Request** – bad request, e.g. unknown filter used
- **401 Unauthorized** – authentication/authorization failed

**GET** `/api/v1/comics/(int: comic_id) /`  
Show a specific comic looked up by comic ID.

#### Example request

```
GET /api/v1/comics/18/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

#### Example response

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  active: true,
  added: "0001-01-01T00:00:00+00:00",
  end_date: null,
  id: "18",
  language: "en",
  name: "xkcd",
  resource_uri: "/api/v1/comics/18/",
  rights: "Randall Munroe, CC BY-NC 2.5",
  slug: "xkcd",
  start_date: "2005-05-29",
  url: "http://www.xkcd.com/"
}
```

#### Parameters

- **comic\_id** – the comic ID

#### Status Codes

- **200 OK** – no error
- **401 Unauthorized** – authentication/authorization failed
- **404 Not Found** – comic not found

### Releases resource

**GET** `/api/v1/releases/`  
Lists all available releases, last fetched first. Supports *Pagination*.

#### Example request

```
GET /api/v1/releases/?comic__slug=xkcd&limit=2 HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

### Example response

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  meta: {
    limit: 2,
    next: "/api/v1/releases/?limit=2&key=76acdcdf16ae4e12becb00d09a9d9456&
↪format=json&comic__slug=xkcd&offset=2",
    offset: 0,
    previous: null,
    total_count: 1104
  },
  objects: [
    {
      comic: "/api/v1/comics/18/",
      fetched: "2012-10-08T04:03:56.411028+00:00",
      id: "147708",
      images: [
        {
          checksum:
↪"605d9a6d415676a21ee286fe2b369f58db62c397bfdfa18710b96dcbcc4df12",
          fetched: "2012-10-08T04:03:56.406586+00:00",
          file: "https://static.example.com/media/xkcd/6/
↪605d9a6d415676a21ee286fe2b369f58db62c397bfdfa18710b96dcbcc4df12.png",
          height: 365,
          id: "151937",
          resource_uri: "/api/v1/images/151937/",
          text: "Facebook, Apple, and Google all got away with their
↪monopolist power grabs because they don't have any 'S's in their names for
↪critics to snarkily replace with '$'s.",
          title: "Microsoft",
          width: 278
        }
      ],
      pub_date: "2012-10-08",
      resource_uri: "/api/v1/releases/147708/"
    },
    {
      comic: "/api/v1/comics/18/",
      fetched: "2012-10-05T05:03:33.744355+00:00",
      id: "147172",
      images: [
        {
          checksum:
↪"6d1b67d3dc00d362ddb5b5e8f1c3f174926d2998ca497e8737ff8b74e7100997",
          fetched: "2012-10-05T05:03:33.737231+00:00",
          file: "https://static.example.com/media/xkcd/6/
↪6d1b67d3dc00d362ddb5b5e8f1c3f174926d2998ca497e8737ff8b74e7100997.png",
          height: 254,
          id: "151394",
```

(continues on next page)

(continued from previous page)

```

        resource_uri: "/api/v1/images/151394/",
        text: "According to my mom, my first word was (looking up at_
→the sky) 'Wow!'",
        title: "My Sky",
        width: 713
    }
],
pub_date: "2012-10-05",
resource_uri: "/api/v1/releases/147172/"
}
]
}

```

### Query Parameters

- **subscribed** – only include releases the authorized user is subscribed to if `true`, or unsubscribed to if `false`
- **comic** – only include releases with matching comic. All filters on the comic resource may be used, e.g. `comic__slug=xkcd`.
- **images** – only include releases with matching image. All filters on the image resource may be used, e.g. `images__height__gt=1000`.
- **pub\_date** – only include releases with `pub_date` matching. Date range queries, like `pub_date__year=2011` or `pub_date__gte=2011-01-01&pub_date__lte=2011-12-31`, are supported.
- **fetches** – only include releases with fetched matching. Date range queries are supported.

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, e.g. unknown filter used
- 401 Unauthorized – authentication/authorization failed

**GET** `/api/v1/releases/(int: release_id)/`  
Show a specific release looked up by release ID.

#### Example request

```

GET /api/v1/releases/147708/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456

```

#### Example response

```

HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  comic: "/api/v1/comics/18/",
  fetched: "2012-10-08T04:03:56.411028+00:00",
  id: "147708",
  images: [
    {

```

(continues on next page)

(continued from previous page)

```

        checksum:
↪ "605d9a6d415676a21ee286fe2b369f58db62c397bfdfa18710b96dcbbcc4df12",
        fetched: "2012-10-08T04:03:56.406586+00:00",
        file: "https://static.example.com/media/xkcd/6/
↪ 605d9a6d415676a21ee286fe2b369f58db62c397bfdfa18710b96dcbbcc4df12.png",
        height: 365,
        id: "151937",
        resource_uri: "/api/v1/images/151937/",
        text: "Facebook, Apple, and Google all got away with their monopolist_
↪ power grabs because they don't have any 'S's in their names for critics to_
↪ snarkily replace with '$'s.",
        title: "Microsoft",
        width: 278
    }
],
pub_date: "2012-10-08",
resource_uri: "/api/v1/releases/147708/"
}

```

### Parameters

- **release\_id** – the release ID

### Status Codes

- 200 OK – no error
- 401 Unauthorized – authentication/authorization failed
- 404 Not Found – release not found

## Images resource

You will probably not use the images resource, as the images are available through the `releases` resource as well. The images resource is included to give the images referenced to by releases their own canonical URLs.

### GET /api/v1/images/

Lists all images. Supports *Pagination*.

#### Query Parameters

- **fetched** – only include images with fetched matching. Date range queries are supported.
- **title** – only include images with matching title. Queries like `title__icontains=cake` are supported.
- **text** – only include images with matching text. Queries like `text__icontains=lies` are supported.
- **height** – only include images with height matching. Integer range queries, like `height__gt=1000` are supported.
- **width** – only include images with width matching. Integer range queries are supported.

#### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, e.g. unknown filter used
- 401 Unauthorized – authentication/authorization failed

**GET** `/api/v1/images/(int: image_id)/`  
 Show a specific image looked up by image ID.

#### Parameters

- **image\_id** – the image ID

#### Status Codes

- **200 OK** – no error
- **401 Unauthorized** – authentication/authorization failed
- **404 Not Found** – image not found

## Subscriptions resource

**GET** `/api/v1/subscriptions/`  
 List all the authenticated user's comic subscriptions. Supports *Pagination*.

#### Example request

```
GET /api/v1/subscriptions/?comic__slug=xkcd HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

#### Example response

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  meta: {
    limit: 20,
    next: null,
    offset: 0,
    previous: null,
    total_count: 1
  },
  objects: [
    {
      comic: "/api/v1/comics/18/",
      id: "2",
      resource_uri: "/api/v1/subscriptions/2/"
    }
  ]
}
```

#### Query Parameters

- **comic** – only include releases with matching comic. All filters on the comic resource may be used, e.g. `comic__slug=xkcd`.

#### Status Codes

- **200 OK** – no error
- **401 Unauthorized** – authentication/authorization failed

**POST /api/v1/subscriptions/**

Subscribe the authenticated user to the given comic.

**Example request**

Note that the request specifies the Content-Type since it includes a body with JSON.

```
POST /api/v1/subscriptions/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
Content-Type: application/json

{
  "comic": "/api/v1/comics/18/"
}
```

**Example response**

```
HTTP/1.0 201 CREATED
Content-Type: text/html; charset=utf-8
Location: https://example.com/api/v1/subscriptions/4/
```

**Status Codes**

- 201 Created – no error, object was created, see Location header
- 401 Unauthorized – authentication/authorization failed
- 500 Internal Server Error – if the request cannot be processed, e.g. because the subscription already exists

**PATCH /api/v1/subscriptions/**

Do bulk updates of subscriptions: e.g. create and delete multiple subscriptions with a single request.

If any part of the bulk update fails, all changes are rolled back.

**Example request**

```
PATCH /api/v1/subscriptions/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
Content-Type: application/json

{
  "objects": [
    {
      "comic": "/api/v1/comics/19/"
    },
    {
      "comic": "/api/v1/comics/20/"
    }
  ],
  "deleted_objects": [
    "/api/v1/subscriptions/4/",
    "/api/v1/subscriptions/5/"
  ]
}
```

**Example response**

```
HTTP/1.0 202 ACCEPTED
Content-Length: 0
Content-Type: text/html; charset=utf-8
```

**Status Codes**

- **202 Accepted** – no error, changes was accepted, use **GET** to see the changes
- **401 Unauthorized** – authentication/authorization failed
- **500 Internal Server Error** – if the request cannot be processed, e.g. because a subscription already exists

**GET** `/api/v1/subscriptions/(int: subscription_id) /`

Show one of the authenticated user's comic subscriptions looked up by subscription ID.

**Example request**

```
GET /api/v1/subscriptions/2/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

**Example response**

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8

{
  comic: "/api/v1/comics/18/",
  id: "2",
  resource_uri: "/api/v1/subscriptions/2/"
}
```

**Parameters**

- **subscription\_id** – the subscription ID

**Status Codes**

- **200 OK** – no error
- **401 Unauthorized** – authentication/authorization failed
- **404 Not Found** – subscription not found

**DELETE** `/api/v1/subscriptions/(int: subscription_id) /`

Unsubscribe the authenticated user from the given comic.

**Example request**

```
DELETE /api/v1/subscriptions/17/ HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Key 76acdcdf16ae4e12becb00d09a9d9456
```

**Example response**

```
HTTP/1.0 204 NO CONTENT
Content-Length: 0
Content-Type: text/html; charset=utf-8
```

### Parameters

- `subscription_id` – the subscription ID

### Status Codes

- `204 No Content` – no error, and no content returned
- `401 Unauthorized` – authentication/authorization failed
- `404 Not Found` – subscription not found

## 1.5 Development

*comics* development is coordinated through [GitHub](#).

### 1.5.1 How to contribute

The easiest way to contribute to *comics* is to register as a user at [GitHub](#), fork [the comics project](#), and start hacking. To get your changes back into *comics*' mainline, send a pull request to [jodal at GitHub](#). Patches accompanied by tests and documentation gives +5 karma and kudos. When hacking on *comics*, please follow the code style and commit guidelines below.

All contributions must be granted under the same license as *comics* itself.

#### Code style

- Follow [PEP 8](#) unless otherwise noted. `pep8.py` can be used to check your code against the guidelines, however remember that matching the style of the surrounding code is also important.
- Use four spaces for indentation, *never* tabs.
- Use CamelCase with initial caps for class names:

```
ClassNameWithCamelCase
```

- Use underscore to split variable, function and method names for readability. Don't use CamelCase.

```
lower_case_with_underscores
```

- Use the fact that empty strings, lists and tuples are False and don't compare boolean values using `==` and `!=`.
- Follow whitespace rules as described in [PEP 8](#). Good examples:

```
spam(ham[1], {eggs: 2})
spam(1)
dict['key'] = list[index]
```

- Limit lines to 80 characters and avoid trailing whitespace. However note that wrapped lines should be *one* indentation level in from level above, except for `if`, `for`, `with`, and `while` lines which should have two levels of indentation:



```
if (foo and bar ...
    baz and foobar):
    a = 1

from foobar import (foo, bar, ...
                    baz)
```

- For consistency, prefer ' over " for strings, unless the string contains ' .
- Take a look at [PEP 20](#) for a nice peek into a general mindset useful for Python coding.

### Commit guidelines

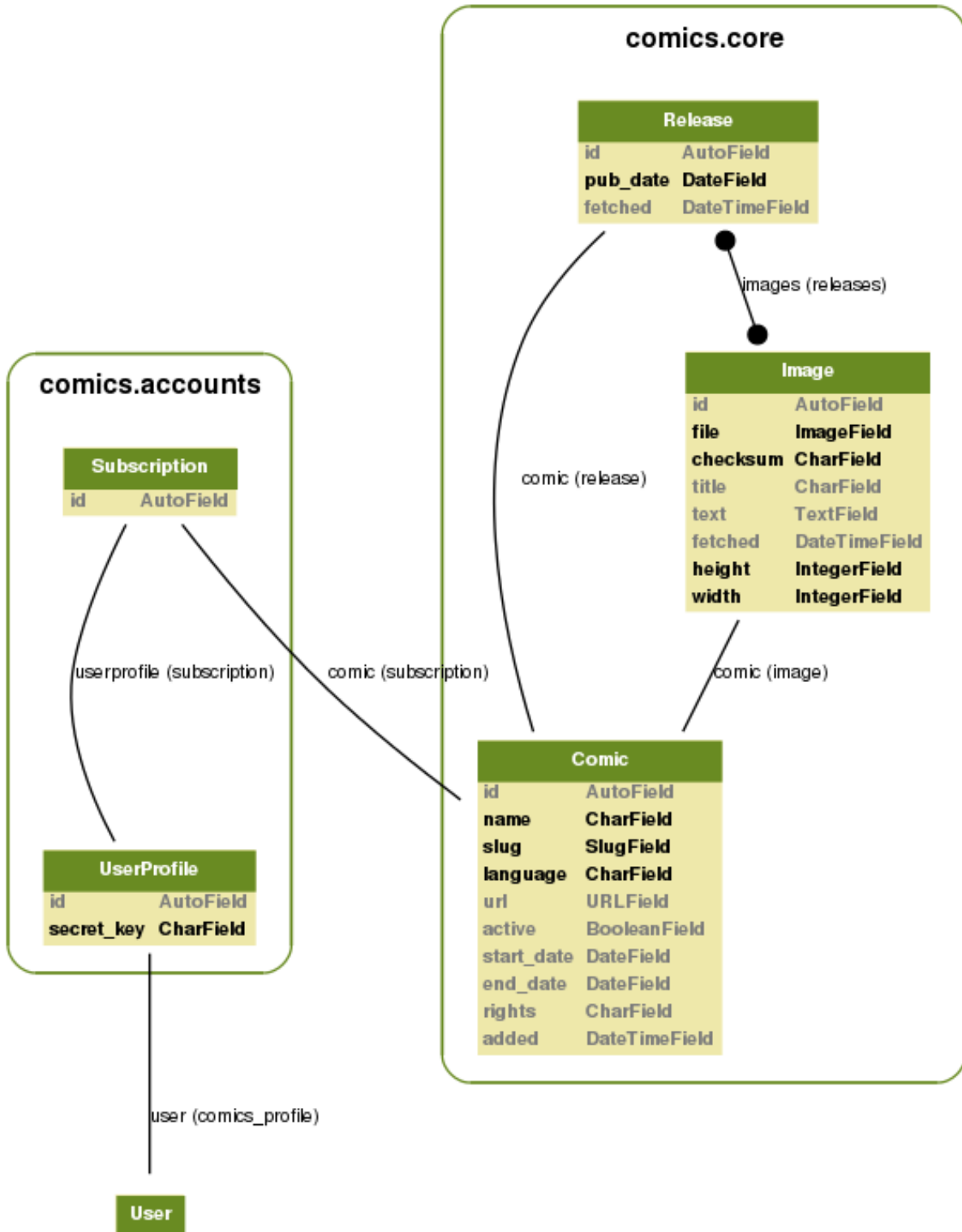
- Keep commits small and on topic, e.g. add one crawler per commit.
- Merge feature branches with `--no-ff` to keep track of the merge.
- When expanding API to accommodate new crawler features, commit API changes, then new crawlers in a separate commit.
- When changing existing API, commit API change and crawler changes in same commit. If this commit looks too big you should be working in a feature branch not a single commit.
- Same policy applies for non-crawler changes.

### 1.5.2 Data model

*comics*' data model is very simple. The `comics.core` app consists of three models; `Comic`, `Release`, and `Image`. The `comics.accounts` app adds a `UserProfile` which add comic specific fields to Django's user model, including a mapping from the user to her preferred comics.

Deprecated since version 2.0: The `comics.sets` app has a model named `Set`. This model is deprecated and replaced by `UserProfile`.

Changes to the data model are managed using Django's built-in database migrations. If you need to change the models, please provide the needed migrations.



The above data model diagram was generated using the Django app `django_extensions` and the following command:

```
python manage.py graph_models --settings=comics.settings.dev \
```

(continues on next page)

(continued from previous page)

```
--output=docs/_static/data_model.png --group-models core accounts
```

### 1.5.3 Running tests

*comics* got some tests, but far from full test coverage. If you write new or improved tests for *comics*' functionality it will be greatly appreciated.

To run unit tests:

```
python manage.py test
```

## 1.6 Authors

*comics* is copyright 2009-2014 Stein Magnus Jodal and contributors. *comics* is licensed under the GNU Affero General Public License version 3.

The following persons have contributed to *comics*. The list is in the order of first contribution. For details on who have contributed what, please refer to our Git repository.

- Stein Magnus Jodal <stein.magnus@jodal.no>
- Thomas Adamcik <adamcik@samfundet.no>
- Johannes Knutsen <johannes@knutseninfo.no>
- Kristian Klette <klette@klette.us>
- Jim Wyllie <jwyllie83@gmail.com>
- Rune Hammersland <rune.hammersland+github@gmail.com>
- Anders Birkenes <datagutten@datagutten.net>
- Jim Frode Hoff <jimfrodehoff@gmail.com>
- Nina Margrethe Smørsgård <nina.m.smorsgard@gmail.com>
- Morten Minde Neergaard <xim-git@akuma.no>
- Torkild Retvedt <torkild@retvedt.no>

## 1.7 Changes

This change log is used to track all major changes to *comics* after the first versioned release.

### 1.7.1 v2.4.3 (2017-04-01)

- Update all crawlers based on `HeltNormaltCrawlerBase` after site change.

#### Crawlers

- New: `businesscat`
- New: `donthitsave`

- New: firekanta
- New: mollybeans
- New: truthfactsgo
- New: wyrd
- Update: lunchtu after site change.
- Update: thegamercat after feed change.

### **1.7.2 v2.4.2 (2016-01-19)**

- Get larger images for all GoComics.com comics.

#### **Crawlers**

- New: bloomcounty2015
- New: marydeath
- New: sarahcandersen
- New: thedanemen
- New: unsounded
- Update: hjalmar to fix crash.
- Update: leasticoulddo to fix image format.
- Update: mutts to use new domain.
- Update: All HeltNormalt.no comic crawlers to set a User-Agent header.

### **1.7.3 v2.4.1 (2015-09-04)**

- Fix references to comics list “below”. It was moved to its own page in v2.3.0.

#### **Crawlers**

- New: criticalmiss
- New: dumbingofage
- New: fowllanguage
- New: jesusandmo
- New: ssss
- Update: adam4d after site change.
- Update: asofterworld after site change.
- Update: babyblues after site change.
- Update: beetlebailey after site change.
- Update: blasternation after feed change.
- Update: cyanideandhappiness after site change.
- Update: dilbert after site change.
- Update: dustin after site change.

- Update: `hijinksensue` after feed change.
- Update: `hjalmar` after file format change.
- Update: `joyoftech` after feed change.
- Update: `kiwiblit` after site change.
- Update: `lizclimo` after feed change.
- Update: `lookingforgroup` after feed change.
- Update: `nerfnow` to include text.
- Update: `pidjin` after feed change.
- Update: `satw` after site change.
- Update: `scenesfromamultiverse` to include User-Agent in request.
- Update: `smbc` after feed change.
- Update: `thegamercat` after feed change.
- Update: `threepanelsoul` after feed change.
- Update: `wumovg` after file format change.
- Update: `yafgc` after feed change.

#### 1.7.4 v2.4.0 (2015-01-03)

- Dependencies with new minimum versions:
  - Django  $\geq 1.7, < 1.8$
  - `django-tastypie`  $\geq 0.12, < 0.13$
- Dependencies with new maximum versions:
  - Pillow  $\geq 1.7, < 2.8$
- Removed dependencies:
  - South
- Switched the database migrations from using South to the builtin tool in Django 1.7. Due to this, all old database migrations have been thrown away. If you're running an old version of *comics*, please upgrade to the latest v2.3.x release first to get your database entirely up to date, then upgrade to v2.4.x. There are no database changes between v2.3.x and v2.4.x.

#### 1.7.5 v2.3.4 (2015-01-03)

##### Crawlers

- New: `awkwardzombie`
- New: `fatawesome`
- New: `lizclimo`
- New: `lunarbaboon`
- Update `zits` to fetch from better source.
- Update `nerfnow` after feed change.

## 1.7.6 v2.3.3 (2014-10-06)

### Crawlers

- New: nerdrage
- Update lots of comic release schedules.
- Update: drmcninja after feed change.
- Update: harkavagrant after site change.
- Update: hijinksensue to fetch large images.
- Update: lookingforgroup to not fetch unrelated comics.
- Update: nedroid after feed change.
- Update: pennyarcade to work around User-Agent check.
- Update: pidjin to ignore repeated image.
- Update: redmeat after site change.
- Update: subnormality after page change.
- Update: wondermark after feed change.
- Inactive: marriedtothesea
- Inactive: pelsogpoter
- Inactive: threewordphrase
- Inactive: toothpastefordinner
- Inactive: virtualshackles

## 1.7.7 v2.3.2 (2014-09-11)

- Dependencies with new maximum versions:
  - django\_compressor  $\geq 1.1, < 1.5$
  - Pillow  $\geq 1.7, < 2.6$
- Add crawler status page to the menu.
- Fix position of “Add to my comics” button.
- Make an attempt at fixing downloading of image URLs with non-ASCII chars in the URL.

### Crawlers

- New: blasternation
- New: cardboardcrack
- New: commitstrip
- New: lunche24
- New: q2qcomics
- Update: asofterworld after site change.
- Update: doghouse after feed change.

- Update: joyoftech after site change.
- Update: kiwiblitiz after site change.
- Update: seemikedraw after site move.
- Update: thegamercat after feed change.
- Update: walkoflife is no longer published.

### 1.7.8 v2.3.1 (2014-06-11)

- Display error message instead of crashing if a password reset link is reused.

#### Crawlers

- New: iamarg
- Update: asofterworld after feed breakage.
- Update: crookedgremlins after feed removal.
- Update: darklegacy after site change.
- Update: hijinksensue after feed change.
- Update: lookingforgroup after feed change.
- Update: mutts after site change.
- Update: phd after feed change.
- Update: poledancingadventures after site change.
- Update: pcweenies after feed change.
- Update: playervsplayer after site change.
- Update: sinfest after site change.
- Update: stickydillybuns after delayed feed update.
- Update: yamac after site change.
- Inactive: antics
- Inactive: boxerhockey
- Inactive: dungeond
- Inactive: eatthattoast
- Inactive: gregcomic
- Inactive: lunch
- Inactive: pinkparts
- Inactive: somethingofthatilk
- Inactive: thechalkboardmanifesto

### 1.7.9 v2.3.0 (2014-04-07)

- Dependencies with new minimum versions:
  - Django  $\geq 1.6$ ,  $< 1.7$
  - django-bootstrap-form  $\geq 3.1$ ,  $< 3.2$
- Dependencies with new maximum versions:
  - cssmin  $\geq 0.1$ ,  $< 0.3$
  - cssselector  $\geq 0.8$ ,  $< 0.10$
  - django-tastypie  $\geq 0.9.13$ ,  $< 0.12$
  - Pillow  $\geq 1.7$ ,  $< 2.5$
- Upgraded to Bootstrap 3, which gives way for a refreshed and more responsive/mobile friendly design.
- Upgraded to Font Awesome 4.
- Upgraded to JQuery 2. IE6/7/8 are no longer supported. IE9 is the oldest IE version you can expect to work with *comics*.
- Moved comics list from bottom of each page to its own page in the top menu. This makes it more available on mobile clients, as well as to new users that don't know where to find the list after they've subscribed to their first comic.
- Use Moment.js to show time since fetched timestamp in release meta data. This can't be done on the server side due to heavy caching.

### 1.7.10 v2.2.3 (2014-03-31)

- Make `num_releases_since/:id` view return 404 instead of 500 for unknown release IDs.

#### Crawlers

- New: adam4d
- New: poorlydrawnlines
- Update: lookingforgroup after feed change.
- Update: pennyarcade after site change.
- Update: questionablecontent after site change.
- Update: satw to include description text.

### 1.7.11 v2.2.2 (2013-12-21)

#### Crawlers

- Update many comic schedules
- Update: hijinksensue after feed change.
- Update: partiallyclips to save larger image.
- Update: scenesfromamultiverse after feed change.
- Update: toothpastefordinner after feed change.



### 1.7.12 v2.2.1 (2013-11-08)

- **Security:** Disabled the GZip middleware to help prevent the BREACH attack. See <https://www.djangoproject.com/weblog/2013/aug/06/breach-and-django/> for details.
- The feedback form no longer uses the logged in user's email address as the sender address as this can cause the mail to be rejected due to sender validation and similar anti spam measures. The user's email is still in the email signature and is now also in the mail's Reply-To header.
- The status page now sorts comics by the number of days since the last release, moving the comics most in need of maintenance to the top of the page.

#### Crawlers

- New: pinkparts
- New: poledancingadventures
- New: redmeat
- New: seemikedraw
- Update: amazingsuperpowers after feed change.
- Update: axecop after feed change.
- Update: bugcomic after site change.
- Update: chainsawsuit after feed change.
- Update: crookedgremlins after site change.
- Update: cyanideandhappiness schedule.
- Update: evilinc after site change.
- Update: fanboys after feed change.
- Update: gregcomic after site change.
- Update: gucomics after feed change.
- Update: harkavagrant after feed change.
- Update: heijibits after addition of User-Agent check.
- Update: hjinksensue after feed change.
- Update: joyoftech after feed change.
- Update: icanbarelydraw after addition of User-Agent check.
- Update: kalscartoon after addition of User-Agent check.
- Update: kiwiblitz after feed change.
- Update: lunch to use feed.
- Update: marriedtothesea after feed change.
- Update: menagea3 after site change and delayed feed update.
- Update: mysticrevolution after site change.
- Update: nedroid after feed change.
- Update: optipess to add related text.
- Update: pidjin after site change.

- Update: questionablecontent after site change.
- Update: reallife after feed change.
- Update: sheldon after site change.
- Update: shortpacked after site change.
- Update: stickydillybuns to use site feed and include title.
- Update: thegamercat after addition of User-Agent check.
- Update: thegutters after feed change.
- Update: wulffmorgenthaler after site change.
- Inactive: picturesforsadchildren
- Inactive: radiogaga
- Inactive: reveland

### 1.7.13 v2.2.0 (2013-07-07)

- New dependencies, and dependencies with new minimum versions:
  - cssselect  $\geq 0.8$ ,  $< 0.9$
  - Django  $\geq 1.5$ ,  $< 1.6$
  - django-tastypie  $\geq 0.9.13$ ,  $< 0.10$
  - lxml  $\geq 3$ ,  $< 4$
  - defusedxml  $\geq 0.4$ ,  $< 0.5$
- Dependencies with new maximum versions:
  - django\_compressor  $\geq 1.1$ ,  $< 1.4$
  - Pillow  $\geq 1.7$ ,  $< 2.2$
  - South  $\geq 0.7$ ,  $< 2.0$
- Fix crash in importing of old comic sets, which has been broken since v2.1.0.

#### Crawlers

- New: 20px
- New: completelyseriouscomics
- New: hjalmar
- New: kollektivet
- New: tommyogtigern
- New: truthfacts
- New: wumovg
- Update: amazingsuperpowers after feed change.
- Update: antics after feed change.
- Update: beyondthetree is no longer published.
- Update: chainsawsuit after feed change.

- Update: choppingblock is no longer published.
- Update: darylccagle after feed change.
- Update: dilbert after removal of feed.
- Update: dilbertvg after move to new site.
- Update: eatthattoast to not throw exception when the site is broken.
- Update: eon is no longer published.
- Update: evilinc to be more robust.
- Update: exiern to track new storyline.
- Update: extralife after feed change.
- Update: extraordinary after site change.
- Update: fagprat after site change.
- Update: geekandpoke after feed change.
- Update: gws after site change.
- Update: harkavagrant schedule.
- Update: havet is no longer published.
- Update: hejibits schedule.
- Update: heltnils is no longer published.
- Update: hipsterhitler is no longer published.
- Update: kiwiblitiz schedule.
- Update: kukuburi is no longer published.
- Update: leasticoulddo after site change.
- Update: manalanextdoor is no longer published.
- Update: manlyguys after feed change.
- Update: orneryboy is no longer published.
- Update: overcompensating is no longer published.
- Update: perrybiblefellowship to be an active crawler again.
- Update: picturesforsadchildren is no longer published.
- Update: pidjin to ignore repeated non-comic image.
- Update: reallife after site change.
- Update: sheldon schedule.
- Update: slagoon after site change.
- Update: smbc after site change.
- Update: somethingofthatilk schedule.
- Update: subnormality to include title text.
- Update: thechalkboardmanifesto schedule.
- Update: thegamercat after feed change.

- Update: theidlestate is no longer published.
- Update: undeclaredmajor is no longer published.
- Update: utensokker is published again.
- Update: uvod after feed change.
- Update: veslemoy is no longer published.
- Update: whiteninja is no longer published.
- Update: wulffmorgenthaler to work after site change.
- Update: wulffmorgenthalerap is no longer active.
- Update: yehudamoon is no longer published.
- Update: zofiesverden is no longer published.

### 1.7.14 v2.1.1 (2013-02-26)

#### Crawlers

- New: lunchtu
- New: mutts. Contributed by Anders Birkenes.
- New: pelsogpoter. Contributed by Anders Birkenes.
- New: stickygillybuns
- New: undeclaredmajor
- New: yamac
- Update: abstrusegoose after feed change.
- Update: bizarro after feed change.
- Update: joyoftech after site change.
- Update: lookingforgroup after feed change.
- Update: thegamercat to be more robust.

### 1.7.15 v2.1.0 (2012-10-15)

- Added a *Web service* interface to the *comics* instance's data to enable the development of custom frontends to *comics* and apps for Android and iOS.
- Improved admin interface. A lot of fields on the comic, release, and image models are now read-only, as they are only intended to be changed by the `comics_addcomics` and `comics_getreleases` commands. The *comics* admin interface is mainly intended for browsing and deleting comics/releases/images, not changing.
- Proper time zone support for comics crawling. We now calculate the current date at the location a comic is published using time zone aware datetime objects for the current time, which are converted to the comic's local time zone using `pytz`.
- Removed the setting `COMICS_DEFAULT_TIME_ZONE`.
- Updated time zone data for all crawlers. A lot of releases will now be fetched an hour earlier during daylight savings time, which is now taken into consideration when crawling.

### 1.7.16 v2.0.1 (2012-10-06)

- Add dependency on `pytz`.
- Make conversion from publication date to epoch used by 11 crawlers aware of the time zone.
- Set sender of feedback emails to the email address of the logged in user.

#### Crawlers

- New: `antics`
- New: `beetlebailey`
- New: `choppingblock`
- New: `dungeond`
- New: `dustin`
- New: `exiern`
- New: `pickles`
- Update: `boxerhockey` after site change.
- Update: `exiern` after site change.
- Update: `gregcomic` schedule.
- Update: `havet` with better time zone handling.
- Update: `kiwiblit` after site change.
- Update: `misfile` after site change.
- Update: `mysticrevolution` to be more robust.
- Update: `reveland` with better time zone handling.
- Update: `spikedmath` to only fetch the correct images.
- Update: `tehgladiators` schedule.
- Update: `thegamercat` to fetch full size images.
- Update: `virtualshackles` schedule.
- Update: `walkoflife` with better time zone handling.
- Update: `whattheduck` schedule.
- Update: `whiteninja` schedule.
- Update: `wulffmorgenthaler` to fetch the previous day due to releases being delayed.
- Update: `yehudamoon` after site change.
- Update: `zelda` schedule.
- Update: `zits` after site change.

### 1.7.17 v2.0.0 (2012-06-11)

Version 2 refreshes the entire *comics* web interface. The aggregation part of *comics* is mostly unchanged since v1.1.

- Design: New design based on Twitter Bootstrap.

- User accounts:
  - Add user account registration flow, which includes email address verification, login, logout, password change, and password reset.
  - Add account management interface.
  - Add user information to footer of emails sent from the feedback page.
  - Require a user specific secret key to allow access to feeds. (Fixes: #25)
  - Add support for requiring an invitation to register as a new user. Set the setting `INVITE_MODE` to `True` to require invitation before registration. (Fixes: #29)
- “My comics”:
  - Replace named comic sets with comic subscriptions associated with users, called “my comics”. An importer for converting old comics sets to “my comics” is included. (Fixes: #26, #27)
  - Add buttons to all comic views for adding the comic to “my comics”.
  - Extend comics list in the footer to include subscription management. (Fixes: #28, #49)
- Comics browsing:
  - Orders the “latest” view by fetched time instead of comic name. New content is always at the top. (Fixes: #13)
  - Removes browsing of weeks or N days, with the exception of +1 days, which is kept as a “today” view.
  - Reimplemented lots of crusty old code using Django’s class-based generic views.
  - Reimplement feeds using regular views instead of Django’s feed abstraction to reduce the feed response time enough to not cause timeouts when using e.g. Netvibes to subscribe to feeds. (Fixes: #5)
- Comics crawling:
  - Try to verify that image files are valid by loading them with PIL before saving them. (Fixes: #17)
  - Use PIL instead of server provided MIME types to identify the image type.
  - Removed unused `check_image_mime_type` crawler setting.
  - Whitelist GIF, JPEG, and PNG files. All other file types are rejected. (Fixes: #16)
  - Blacklisted the GoComics placeholder image.
- Development:
  - The WSGI file is now also used when using Django’s `runserver` command while developing, making the development and deployment environments more alike.

### v1.1 to v2.0 migration guide

- New dependencies:
  - `django-registration` `>= 0.8, < 0.9`
  - `django-bootstrap-form` `>= 2.0, < 2.1`
- Updated dependencies:
  - Django `>= 1.4, < 1.5`
  - `django_compressor` `>= 1.1, < 1.2`
- Settings:

- Removed setting `COMICS_SITE_TAGLINE`.
- Replaced setting `COMICS_MAX_DAYS_IN_PAGE` with `COMICS_MAX_RELEASES_PER_PAGE`.
- Removed `COMICS_MEDIA_ROOT` and `COMICS_MEDIA_URL`. As static files now are located under `STATIC_ROOT` and `STATIC_URL`, the entire namespace under `MEDIA_ROOT` and `MEDIA_URL` are now available for downloaded media, e.g. crawled comics.

- **Commands:**

- `loadmeta` is now called `comics_addcomics`. It no longer defaults to adding all comics to your installation, but you must now specify `-c all` to do so.
- `getcomics` is now called `comics_getreleases`

Remember to update your cronjobs.

- **Project layout:**

- Moved `manage.py` one level higher in the directory structure, to follow the new defaults in Django 1.4. Again, remember to update your cronjobs.
- Moved file with WSGI application from `wsgi/deploy.wsgi` to `comics/wsgi/__init__.py` to follow the new default structure in Django 1.4. Remember to update your web server configuration.

- As the comic sets functionality have been replaced, the app `comics.sets` is no longer activated by default. If you're upgrading from comics v1.x and have existing sets in your database, you *should* activate the `comics.sets` app so that your users may import their old comic sets into their new user accounts. Add the following to your local settings file, `comics/settings/local.py`:

```
from comics.settings.base import INSTALLED_APPS
INSTALLED_APPS += ('comics.sets',)
```

- Renamed `MetaBase` to `ComicDataBase`, and moved it to `comics.core.comic_data`. Remember to update any custom crawlers.

- **Database changes:**

- The field `Comic.number_of_sets` have been removed as it is no longer used. If you would want to rollback from 2.x to 1.x the data in this field can be regenerated, as it's only a denormalization of data available elsewhere.
- The datetime field `Comic.added` has been added. It is automatically populated with a date in the far past upon database migration.
- Added two new database indexes to the `Release` model, which both help a lot towards making comics browsing faster. They will be automatically created on database migration.

All of these changes can be automatically applied to your database. To do so, run:

```
python manage.py syncdb --migrate
```

## 1.7.18 v1.1.6 (2012-06-10)

### Bugfixes

- `LxmlParser.text()` now returns an empty list if `allow_multiple` is `True` and default is not specified. This is identical to how all other `LxmlParser` selector methods already work.

### Crawlers

- New: `oatmeal`

- New: zelda
- Update: abstrusegoose has a schedule.
- Update: apokalips is no longer published.
- Update: asofterworld after feed change.
- Update: atheistcartoons is no longer published.
- Update: axecop has a schedule.
- Update: basicinstructions has a new schedule.
- Update: bgobt is no longer published.
- Update: boasas is no longer published.
- Update: bunny is no longer published.
- Update: carpediem is no longer published.
- Update: countyoursheep is no longer published.
- Update: crfh after site change.
- Update: darklegacy does not follow a schedule.
- Update: devilbear does not follow a schedule.
- Update: dieselsweetiesweb to be more robust to missing elements in the feed.
- Update: goblins does not follow a schedule.
- Update: gunshow has a new release schedule.
- Update: hijinksensue after feed change.
- Update: icanbarelydraw has a new release schedule.
- Update: kiwiblitiz does not follow a schedule.
- Update: littlegamers does not follow a schedule.
- Update: m is no longer published.
- Update: magpieluck is no longer published.
- Update: pcweenies does not follow a schedule.
- Update: picturesforsadchildren after feed change.
- Update: radiogaga has a new release schedule.
- Update: rhymeswithwitch is no longer published.
- Update: spaceavalanche after feed change.
- Update: stuffnoonetoldme is no longer published.
- Update: subnormality got a sensible history capability.
- Update: tehgladiators does not follow a schedule.
- Update: theidlestate does not follow a schedule.
- Update: utensokker is no longer published.
- Update: uvod got an updated homepage address.
- Update: virtualshackles does not follow a schedule.



- Update: `walkoflife` does not follow a schedule.

### 1.7.19 v1.1.5 (2012-05-09)

The regular crawler updates and a small bug fix.

#### Bugfixes

- Handle aggregated images with MIME type `image/pjpeg` as JPEG images instead of rejecting them.

#### Crawlers

- New: `chainsawsuit`
- New: `goblins`
- New: `subnormality`
- Update: `applegeeks` was discontinued a long time ago.
- Update: `applegeekslite` was discontinued a long time ago.
- Update: `calamitesofnature` has been discontinued.
- Update: `duelinganalogs` was broken due to feed changes.
- Update: `fagprat` has a new schedule.
- Update: `fanboys` was broken due to feed changes.
- Update: `heltnils` has a new schedule.
- Update: `hijinksensure` was broken due to feed changes.
- Update: `playervsplayer` was broken due to feed changes.
- Update: `pondus` was broken due to a site change.
- Update: `savagechickens` has a new schedule.
- Update: `theidlestate` after site redesign and addition of a feed.
- Update: `veslemoy` has a new schedule.

### 1.7.20 v1.1.4 (2012-04-07)

The regular crawler updates and a performance improvement.

#### Bugfixes

- Store only the name of recently used sets in the session, instead of full set objects. After applying this fix, you should either delete all existing sessions:

```
$ python manage.py shell
>>> from django.contrib.sessions.models import Session
>>> Session.objects.all().delete()
```

Or migrate the content of your existing sessions:

```
$ python manage.py cleanup
$ python manage.py shell

# Then run the following Python script in the Python shell:
```

(continues on next page)

(continued from previous page)

```
from django.contrib.sessions.backends.db import SessionStore
from django.contrib.sessions.models import Session
store = SessionStore()
for session in Session.objects.all():
    data = session.get_decoded()
    set_names = []
    for set in data.get('recent_sets', []):
        if hasattr(set, 'name'):
            set_names.append(set.name)
        else:
            set_names.append(set)
    data['recent_sets'] = set_names
    session.session_data = store.encode(data)
    session.save()
print '.',
```

### Crawlers

- New: kellermannen
- New: manalanextdoor
- New: thegamercat
- New: walkoflife
- Update darylcaagle after feed change.
- Update playervsplayer after feed change.

### 1.7.21 v1.1.3 (2012-01-29)

This release adds 9 new crawlers and updates 46 existing crawlers.

#### Crawlers

- New: beyondthetree
- New: dresdencodak
- New: extraordinary
- New: gunnerkrigg
- New: icanbarelydraw. Contributed by Jim Frode Hoff.
- New: manlyguys. Contributed by Jim Frode Hoff.
- New: menagea3
- New: sequentialarts
- New: somethingofthatilk. Contributed by Jim Frode Hoff.
- Update amazingsuperpowers with new release schedule.
- Update billy which is no longer published.
- Update bizarro with new release schedule.
- Update bizarrono which is no longer published.
- Update boasas after site change.

- Update `bgobt` with new release schedule.
- Update `buttersafe` with new release schedule.
- Update `calvinandhobbes` after site change.
- Update `carpediem` after site change.
- Update `darylcastle` after site change.
- Update `devilbear` with new release schedule.
- Update `eatthatoast` after site change.
- Update `eon` after site change.
- Update `extralife` to be more robust.
- Update `fanboys` after site change.
- Update `gregcomic` with new release schedule.
- Update `gucomics` after site change.
- Update `heltnils` after site change.
- Update `hipsterhitler` after site change.
- Update `kalscartoon` after site change.
- Update `lefthandedtoons` with new release schedule.
- Update `loku` which is no longer published.
- Update `m` with new release schedule.
- Update `mortenm` which is no longer published.
- Update `mysticrevolution` after site change.
- Update `nemibt` with new release schedule.
- Update `nerfnow` with new release schedule.
- Update `optipess` with new release schedule.
- Update `orneryboy` with new release schedule.
- Update `pidjin` after site change.
- Update `pondusno` which is no longer published.
- Update `questionablecontent` to be more robust.
- Update `radiogaga` after site change.
- Update `reallife` with new release schedule.
- Update `reveland` with new release schedule.
- Update `romanticallyapocalyptic` to be more robust.
- Update `savagechickens` with new release schedule.
- Update `sheldon` with new release schedule.
- Update `somethingpositive` after site change.
- Update `stickycomics` after site change.
- Update `tehgladiators` after site change.

- Update `thedreamer` with new release schedule.
- Update `threewordphrase` to be more robust.
- Update `utensokker` with new release schedule.
- Update `wulffmorgenthalerap` after site change.
- Update `yehudamoon` with new release schedule.

### 1.7.22 v1.1.2 (2011-09-18)

A couple of bugfixes easing the transition from 1.0.x to 1.1.x by `jwyllie83`, and some new crawlers.

#### Bugfixes

- Updated South requirement to v0.7, which is needed to support the last migration introduced by comics v1.1.0.
- If you use WSGI, you can now add a file `wsgi/local.py` based off of `wsgi/local.py.template` to set local settings for WSGI, like the use of `virtualenv` and debugging settings. This removes the need for changing Git tracked files, like `deploy.wsgi` for adding e.g. `virtualenv` support.

#### Crawlers

- New: `buttersafe`
- New: `doghouse`
- New: `eatthattoast`
- New: `hejibits`
- New: `optipess`
- New: `savagechickens`
- New: `threewordphrase`
- New: `timetrabble`
- Update `pennyarcade` after site change.

### 1.7.23 v1.1.1 (2011-08-22)

Some fixes a week after the v1.1 feature release.

#### Bugfixes

- Fix missing whitespaces on about page after HTML minification.
- Add missing CSRF token to feedback form.

#### Crawlers

- Update `asfterworld` to work after feed change.

### 1.7.24 v1.1.0 (2011-08-15)

- New/upgraded requirements:
  - Django 1.3
  - `django_compressor`

- cssmin
- jsmin
- Page speed improvements:
  - CSS and JavaScript is now merged and minified.
  - HTML is minified.
  - Optional Google Analytics code is upgraded to the asynchronous version.
  - All icons have been replaced with sprites to reduce number of elements that must be downloaded.
- Slightly refreshed graphical design.
- The “all comics” list at the bottom of the page have been changed from a cloud to lists in four columns.
- The optional comic meta data search functionality have been removed.
- Better handling of inactive comics:
  - Add `active` flag to comics.
  - Marked no longer published comics as inactive.
  - Inactive comics are no longer loaded by the `loadmeta` command unless explicitly specified by name or they have been previously loaded. In other words, inactive comics will not automatically be included in new installations.
  - Inactive comics are no longer included in the top 10 on the front page.
  - Inactive comics are now marked in the comics list on the bottom of all pages.
  - Inactive comics are now marked in the comics list page.
  - Inactive comics are now excluded from the set edit form, effectively removing them from the set on save.

### v1.0.x to v1.1.x migration guide

Ordered steps for synchronizing your v1.0.x installation with v1.1.0. You should perform them in order.

#### Using `virtualenv`

If you choose to use `virtualenv` keeping all of comics’ dependencies sandboxed, be sure to activate the environment both in your cronjob and when manually executing `manage.py`:

```
source <path_to_virtualenv>/bin/activate
python manage.py getcomics
```

If you use WSGI, the WSGI file must be modified to support `virtualenv`. See the source code for how the bundled WSGI file solves this.

#### New dependencies

There are several new dependencies. All of them are listed in the file `requirements.txt` and may be installed using `pip`, optionally inside a `virtualenv`:

```
pip install -r requirements.txt
```

To avoid compiling dependencies which are not pure Python and thus requires the installation of various C libraries and Python’s development packages, it may be wise to use your distribution’s package manger for some packages, like `lxml` and `PIL`. E.g. on Ubuntu I would install the dependencies like this:

```
sudo apt-get install python-lxml python-imaging
pip install -r requirements.txt
```

This way, `lxml` and `PIL` are installed from APT, and `pip` installs the remaining pure Python dependencies.

### Settings changes

Database settings now use the new [Django 1.2 format](#). See `comics/settings/base.py` for the new default setting and use it as an example for porting your `comics/settings/local.py` settings file.

### Database migration

A new database field has been added. To migrate your database to work with v1.1.0, run:

```
python manage.py migrate
```

**Warning:** You need South v0.7 or later to perform the database migration.

`comics v1.1.0's requirements.txt` file only require South v0.6 or later. This is a bug, and the migration will not work if you're using South v0.6.x.

### Static files collection

We now use Django's new static files system. After installing you need to "collect" your static files. See [Collecting static files](#) for how to do this.

## 1.7.25 v1.0.8 (2011-08-10)

Just new and updated crawlers.

### Crawlers

- New: `mysticrevolution`
- New: `theidlestate`
- Update `havet` to work after feed removal.
- Update `reveland` to work after feed removal.
- Update `thechalkboardmanifesto` to work after feed change.
- Update `utensokker` to work after feed removal.
- Update `whattheduck` schedule.

## 1.7.26 v1.0.7 (2011-07-13)

Just new and updated crawlers.

### Crawlers

- New: `fagprat`
- New: `gregcomic`
- New: `satw`
- New: `shortpacked`

- New: `stickycomics`
- New: `tehgladiators`
- Update `betty` which has moved from `comics.com` to `gocomics.com`.
- Update `bizarro` which moved to a new site.
- Update `brandondraws` which is no longer published.
- Update `countyoursheep` after URL changes.
- Update `darylcaagle` after change from GIF to JPEG.
- Update `faktafraverden` which is no longer published.
- Update `fminus` which has moved from `comics.com` to `gocomics.com`.
- Update `getfuzzy` which has moved from `comics.com` to `gocomics.com`.
- Update `lookingforgroup` after feed change.
- Update `m as it` moved from `start.no` to `dagbladet.no`.
- Update `nemibt` to work after site change.
- Update `nerfnow` which crashed when it did not find an image URL.
- Update `peanuts` which has moved from `comics.com` to `gocomics.com`.
- Update `pearlsbeforeswine` which has moved from `comics.com` to `gocomics.com`.
- Update `pondusbt` after URL changes.
- Update `rockybt` to work after site change.
- Update `romanticallyapocalyptic` to use web page instead of feed.
- Update `roseisrose` which has moved from `comics.com` to `gocomics.com`.
- Update `treadingground` to not crash if URL is not found.
- Update `threadingground` which is no longer published.
- Update `yehudamoon` which was broken by addition of new images.
- Update `zits` with new feed URL.
- Update generic GoComics.com crawler to also support larger Sunday issues.

### 1.7.27 v1.0.6 (2011-02-19)

The 1.0.6 release includes two bugfixes, five new crawlers, and many updated crawlers. Also, most crawler schedules have been updated to make the status page more useful.

#### Bugfixes

- `comics.aggregator.xmlparser.LxmlParser` methods now returns an empty list if `allow_multiple` is `True` and no value is given for default. This ensures that using the return value in for loops will not fail if no matches are found.
- `comics.aggregator.crawler.CrawlerBase.get_crawler_release()` does no longer create empty releases if the `do_crawl` method returns false values, like empty lists. It previously only stopped processing if `do_crawl` returned `None`.
- Remove safe filter from title text, and explicitly use `escape`, even though they should be implicitly escaped. Thanks to XKCD #859.

### Crawlers

- A lot of comic release schedule updates.
- New: nerfnow
- New: romanticallyapocalyptic
- New: schlockmercenary
- New: spaceavalanche
- New: treadingground
- Update butternutsquash which is no longer published.
- Update charliehorse which is no longer published.
- Update garfield to include Sunday editions.
- Update hipsterhitler to work after feed change.
- Update idiotcomics which is no longer published.
- Update inktank which is no longer published.
- Update intelsinsides which is no longer published.
- Update kiwiblitiz to work after feed change.
- Update lifewithrippy which is no longer published.
- Update pcweenies to work after feed change.
- Update petpeevey which is no longer published.
- Update smbc to work after feed change.
- Update superpoop which is no longer published.
- Update thegutters to use feed instead of broken page parser.
- Update threepanelsoul to work after feed change.
- Update userfriendly to support reruns.
- Update wulffmorgenthaler to work after site change.

### 1.7.28 v1.0.5 (2010-12-29)

A couple of bugfixes, and new and updated crawlers.

#### Bugfixes

- Do not throw `ParserError` in `comics.aggregator.lxmlparser` when the XML document is a all-space string.
- Catch `socket.error` in `comics.aggregator.downloader`, like we already do in `comics.aggregator.crawler`.

#### Crawlers

- New: brandondraws
- New: crookedgremlins
- New: faktafraverden
- New: lunchdb



- New: `orneryboy`
- New: `reveland`
- Update `foxtrot` crawler to work after site change.
- Update `gws` to work again, and add text parsing.
- Update `havet` meta data.
- Update `lookingforgroup` to ignore non-comic releases and fetch multiple pages released on the same day.
- Update `magpieluck` to handle titles without a dash.
- Update `questionablecontent` to not check if the page contains the expected date, as that make us lose some releases.
- Update `utensokker` to use RSS feed.

### 1.7.29 v1.0.4 (2010-10-23)

Yet another minor release bringing a bug fix, four new and five updated crawlers.

#### Bugfixes

- Catch `socket.error` in `CrawlerBase.get_crawler_release()`.

#### Crawlers

- New: `hipsterhitler`
- New: `marriedtothesea`
- New: `stuffnoonetoldme`
- New: `utensokker`
- Update `boxerhockey` to use feed instead of site.
- Update `bugcomic` to not fail if URL is not empty, and to work after source site changes.
- Update `extralife` to work after source site changes.
- Update `gunshow` to work after source site changes.
- Update `questionablecontent` to use site instead of feed, since it lacks some releases.

### 1.7.30 v1.0.3 (2010-07-26)

Another minor release bringing 17 new and 11 updated crawlers.

#### Bugfixes

- Make crawlers handle `httpplib.BadStatusLine` exception raised when HTTP responses are empty.
- Make crawlers convert `lxml.etree._ElementUnicodeResult` to unicode objects before saving to the database, to avoid `DatabaseError: can't adapt errors`.
- Handle MIME types like `image/jpeg`, `image/jpeg` returned by `mimetools.Message.gettype`.
- Use `Crawler.headers` for image requests, and not just page requests.

#### Crawlers

- New: `apokalips`

- New: `axecop` (fixes GH-8)
- New: `boxerhockey`
- New: `bugcomic` (fixes GH-11)
- New: `carpediem`
- New: `crfh`
- New: `darylcagle`
- New: `havet` (fixes GH-7)
- New: `heltnils`
- New: `intelsinsides` (fixes GH-9)
- New: `misfile` (fixes GH-3)
- New: `notinventedhere` (fixes GH-4)
- New: `pondusno`
- New: `radiogaga`
- New: `scenesfromamultiverse` (fixes GH-10)
- New: `sheldon`
- New: `thegutters`
- Update `8bittheater` which is no longer published.
- Update `brinkerhoff` which is no longer published.
- Update `ctrlaltdel` to work after source site changes.
- Update `ctrlaltdeltesillies` to work after source site changes.
- Update `dieselsweetiesweb` to work after source site changes.
- Update `eon` with new source site.
- Update `lunch` with new source site.
- Update `somethingpositive` to get all releases.
- Update `supereffective` to work after source site changes.
- Update `vocats` to work after source site changes.
- Update `yafgc` to work after source site changes.

### 1.7.31 v1.0.2 (2010-04-11)

A minor release to get crawler updates out there. Two new cool but partly immature features are included, as they do not affect existing features or change database schema.

#### Features

- Add status page which for each comic shows when releases are fetched compared to the comic's release schedule. Contributed by Thomas Adamcik.
- Add support for search in comic's title and text fields, using Haystack. Contributed by Thomas Adamcik.

#### Crawlers

- New: `atheistcartoons`

- New: `petpeevey`
- Update `evilinc` to work again.
- Update `uvod` to fetch comment too.
- Update `gunshow` to fetch multiple images per release.
- Update `questionablecontent` to work again.
- Update `basicinstructions` to ignore QR Code.
- Update `partiallyclips` with new feed URL.
- Update `somethingpositive` with new image URL.
- Update `spikedmath` to fetch multiple images per release.

### 1.7.32 v1.0.1 (2010-02-23)

This release features 17 new crawlers and three updated crawlers, most by *comic*'s new contributor Jim Wyllie. Let's get more of those!

#### Features

- Add links to official sites via redirect page.
- Add `comics.aggregator.crawler.GoComicsComCrawlerBase` for fast `gocomics.com` crawler creation.
- Add `headers` argument to `comics.aggregator.lxmlparser.LxmlParser` for adding HTTP headers to requests it makes.
- Add time since last release to `release-list` and `comic-list`.

#### Crawlers

- New: `babyblues`
- New: `calamitiesofnature`
- New: `charliehorse`
- New: `fminus`
- New: `forbetterorforworse`
- New: `girlgenius`
- New: `hijinksensue`
- New: `joelovescrappymovies`
- New: `magpieluck`
- New: `nonsequitur`
- New: `overcompensating`
- New: `pluggers`
- New: `tankmcnamara`
- New: `theboondocks`
- New: `thedreamer`
- New: `wondermark`

- New: `yehudamoon`
- Update `playervsplayer` to not miss comics on days with blog posts.
- Update `questionablecontent` to include text below image.
- Update `kalscartoon` after target site change.
- Update `butternutsquash` after target site change.

### 1.7.33 v1.0.0 (2010-01-27)

A week has gone, and here is the 1.0.0 final release. Enjoy :-)

#### Crawlers

- Update `uvod` crawler to use new feed.

### 1.7.34 v1.0.0.rc2 (2010-01-19)

Second release candidate for 1.0.0. Again, I will bump to 1.0.0 in a week if no new issues arises.

#### Bugfixes

- Make `core/0006` migration work on the `sqlite3` backend.

### 1.7.35 v1.0.0.rc1 (2010-01-17)

First release, so no list of changes. Will bump to 1.0.0 in a week if no issues arise. Please report any problems at <http://github.com/jodal/comics/issues>.

Development on *comics* as a Python/Django project started in February 2007, so this release has been almost three years in the making. Far too long, and I promise it won't be three years until the next release.

## CHAPTER 2

---

### Indices and tables

---

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



---

## HTTP Routing Table

---

### /api

GET /api/v1/, 19  
GET /api/v1/comics/, 20  
GET /api/v1/comics/(int:comic\_id)/, 21  
GET /api/v1/images/, 24  
GET /api/v1/images/(int:image\_id)/, 24  
GET /api/v1/releases/, 21  
GET /api/v1/releases/(int:release\_id)/,  
23  
GET /api/v1/subscriptions/, 25  
GET /api/v1/subscriptions/(int:subscription\_id)/,  
27  
GET /api/v1/users/, 19  
POST /api/v1/subscriptions/, 25  
DELETE /api/v1/subscriptions/(int:subscription\_id)/,  
27  
PATCH /api/v1/subscriptions/, 26





**C**

`comics.aggregator.feedparser`, 14  
`comics.aggregator.lxmlparser`, 13  
`comics.settings.base`, 7



**A**

ACCOUNT\_ACTIVATION\_DAYS (in module comics.settings.base), 7  
 ACCOUNT\_INVITATION\_DAYS (in module comics.settings.base), 7  
 active (ComicData attribute), 11  
 all() (comics.aggregator.feedparser.FeedParser method), 15  
 alt() (comics.aggregator.lxmlparser.LxmlParser method), 14

**C**

ComicData (built-in class), 10  
 comics.aggregator.feedparser (module), 14  
 comics.aggregator.lxmlparser (module), 13  
 comics.settings.base (module), 7  
 COMICS\_BROWSER\_REFRESH\_INTERVAL (in module comics.settings.base), 7  
 COMICS\_GOOGLE\_ANALYTICS\_CODE (in module comics.settings.base), 7  
 COMICS\_IMAGE\_BLACKLIST (in module comics.settings.base), 7  
 COMICS\_LOG\_FILENAME (in module comics.settings.base), 8  
 COMICS\_MAX\_DAYS\_IN\_FEED (in module comics.settings.base), 8  
 COMICS\_MAX\_RELEASES\_PER\_PAGE (in module comics.settings.base), 8  
 COMICS\_NUM\_DAYS\_COMIC\_IS\_NEW (in module comics.settings.base), 8  
 COMICS\_SITE\_TITLE (in module comics.settings.base), 8  
 content0 (comics.aggregator.feedparser.Entry attribute), 15  
 Crawler (built-in class), 11

**D**

DATABASES (in module comics.settings.base), 8

**E**

end\_date (ComicData attribute), 11  
 Entry (class in comics.aggregator.feedparser), 15

**F**

FeedParser (class in comics.aggregator.feedparser), 15  
 for\_date() (comics.aggregator.feedparser.FeedParser method), 15

**H**

has\_rerun\_releases (Crawler attribute), 11  
 headers (Crawler attribute), 11  
 history\_capable\_date (Crawler attribute), 11  
 history\_capable\_days (Crawler attribute), 11  
 href() (comics.aggregator.lxmlparser.LxmlParser method), 14  
 html() (comics.aggregator.feedparser.Entry method), 15

**I**

id() (comics.aggregator.lxmlparser.LxmlParser method), 14  
 INVITATIONS\_PER\_USER (in module comics.settings.base), 8  
 INVITE\_MODE (in module comics.settings.base), 8

**L**

LxmlParser (class in comics.aggregator.lxmlparser), 13

**M**

MEDIA\_ROOT (in module comics.settings.base), 8  
 MEDIA\_URL (in module comics.settings.base), 8  
 multiple\_releases\_per\_day (Crawler attribute), 11

**N**

name (ComicData attribute), 10

**P**

Python Enhancement Proposals

PEP 20, 29

PEP 8, 28

## R

remove() (comics.aggregator.lxmlparser.LxmlParser method), 14

rights (ComicData attribute), 11

## S

schedule (Crawler attribute), 11

SESSION\_COOKIE\_AGE (in module comics.settings.base), 8

src() (comics.aggregator.lxmlparser.LxmlParser method), 13

start\_date (ComicData attribute), 11

STATIC\_ROOT (in module comics.settings.base), 8

STATIC\_URL (in module comics.settings.base), 8

summary (comics.aggregator.feedparser.Entry attribute), 15

## T

tags (comics.aggregator.feedparser.Entry attribute), 16

text() (comics.aggregator.lxmlparser.LxmlParser method), 13

time\_zone (Crawler attribute), 11

TIME\_ZONE (in module comics.settings.base), 8

title() (comics.aggregator.lxmlparser.LxmlParser method), 14

## U

url (ComicData attribute), 11

## V

value() (comics.aggregator.lxmlparser.LxmlParser method), 14