
feeds Documentation

Release 2022.6.18

Florian Preinstorfer, Lukas Anzinger

Jun 18, 2022

Contents

1	Get Feeds	3
2	Quickstart	5
3	Configure Feeds	7
4	Supported Websites	13
5	Supporting a new Website	29
6	Docker	35
7	API for Spiders	37
8	Contribute	41
9	License	43
10	About Feeds	55
11	Related work	57
12	Authors	59

Feeds provides DIY Atom feeds in times of social media and paywall.

CHAPTER 1

Get Feeds

Feeds is meant to be installed on your server and run periodically in a cron job or similar job scheduler. We recommend to install Feeds inside a virtual environment.

Feeds can be installed from PyPI using `pip`:

```
$ pip install PyFeeds
```

You may also install the current development version. The master branch is considered stable enough for daily use:

```
$ pip install https://github.com/pyfeeds/pyfeeds/archive/master.tar.gz
```

After installation `feeds` is available in your virtual environment.

Feeds supports Python 3.7+.

CHAPTER 2

Quickstart

Feeds has a few commands that are described on this page.

- List all available spiders:

```
$ feeds list
```

- Feeds allows to crawl one or more spiders without a configuration file, e.g.:

```
$ feeds crawl indiehackers.com
```

- A *configuration file* is supported too. Simply copy the *Example configuration* and adjust it. Enable the spiders you are interested in and adjust the `output_path` where Feeds stores the scraped Atom feeds:

```
$ cp feeds.cfg.dist feeds.cfg
$ $EDITOR feeds.cfg
$ feeds --config feeds.cfg crawl
```

- Perform a cache cleanup:

```
$ feeds --config feeds.cfg cleanup
```

- Point your feed reader to the generated Atom feeds and start reading. Feeds works best when run periodically in a cron job or similar job scheduler.
- Run `feeds --help` or `feeds <subcommand> --help` for help and usage details.

Configure Feeds

3.1 Feeds settings

Configuration settings related to Feeds need to be specified within the `[feeds]` section of the configuration file. The following settings are supported.

3.1.1 useragent

The Useragent used for crawling.

```
[feeds]
useragent = feeds (+https://github.com/pyfeeds/pyfeeds)
```

3.1.2 spiders

Each spider listed in the `spiders` setting will be crawled with each run. List one spider per line.

```
[feeds]
spiders =
    tvthek.orf.at
    oel.orf.at
```

Use `feeds list` to get a list of all available spiders.

3.1.3 output_path

This is the path where the generated Atom feeds will be saved. You may serve this directory with any webserver.

```
[feeds]
output_path = output
```

3.1.4 output_url

The URL of the target directory from which the feeds can be accessed. This is an optional setting and it is used to generate `atom:link` element with `rel="self"` attribute. See also: <https://validator.w3.org/feed/docs/warning/MissingSelf.html>

```
[feeds]
output_url = https://example.com/feeds
```

3.1.5 truncate_words

Truncate content to 10 words instead of including the full text. This can be useful if generated feeds should be made publicly available.

```
[feeds]
truncate_words = 10
```

3.1.6 remove_images

Remove images from output. This can be useful if generated feeds should be made publicly available.

```
[feeds]
remove_images = 1
```

3.1.7 cache_enabled

Feeds can be configured to use a cache for HTTP responses which is highly recommended to save bandwidth. The `cache_enabled` setting controls whether caching is used.

```
[feeds]
cache_enabled = 1
```

3.1.8 cache_dir

The path where cache data is stored.

```
[feeds]
cache_dir = ~/.cache/feeds
```

3.1.9 cache_expires

Expire (remove) entries from cache after 90 days.

```
[feeds]
cache_expires = 90
```

3.2 Spider specific settings

Some spiders support additional settings. Head over to the *Supported Websites* section for more information on spider specific settings.

3.3 Example configuration

Have a look at Feeds example configuration when configuring Feeds to suit your needs.

```
# Feeds configuration.

[feeds]
# Useragent to use for crawling.
useragent = feeds (+https://github.com/pyfeeds/pyfeeds)

## List of spiders to run by default, one per line.
# spiders =
#     tvthek.orf.at
#     oel.orf.at

## Target directory where the feeds will be saved.
# output_path = output

## URL of target directory from which the feeds can be accessed.
## Optional; used to generate atom:link element with rel="self" attribute.
## See also: https://validator.w3.org/feed/docs/warning/MissingSelf.html
# output_url = https://example.com/feeds

## Truncate content to 10 words instead of including the full text.
## This can be useful if generated feeds should be made publicly available.
# truncate_words = 10
## Remove images from output.
# remove_images = 1

## Enable caching of responses
# cache_enabled = 1
## Path to the cache.
# cache_dir = ~/.cache/feeds
## Expire (remove) entries from cache after 90 days
# cache_expires = 90

#[generic]
## A list of URLs to RSS/Atom feeds.
# urls =
## A list of URLs to RSS/Atom feeds that provide the full content in the "encoded" or
## "content" tag.
# fulltext_urls =

#[falter.at]
## falter.at has a paywall for certain articles.
## If you want to crawl paid articles, please provide abonr (subscription
## number) and password.
# abonr =
# password =
# blogs =
```

(continues on next page)

(continued from previous page)

```
#      lingens
#      thinktank

#[konsument.at]
## KONSUMENT.AT has a paywall for certain articles.
## If you want to crawl paid articles, please provide username and password.
# username =
# password =

#[biblioweb.at]
## Location of your library that uses biblioweb.at.
# location =

#[lwn.net]
## LWN.net has paywalled articles.
## If you want to crawl them, please provide username and password.
# username =
# password =

#[vice.com]
#locales =
#   de_at
#   de

#[nachrichten.at]
## Nachrichten.at has paywalled articles.
## If you want to crawl them, please provide username and password.
#username =
#password =
#ressorts =
#   wels
#   linz
#   nachrichten

#[uebermedien.de]
## uebermedien.de has a paywall for certain articles.
## If you want to crawl paid articles, please provide your Steady username
## and password.
# username =
# password =

#[orf.at]
#channels =
#   news
#   fm4
#   science
#   help
#   sport
#   oe3
#   oesterreich
#   burgenland
#   wien
#   noe
#   ooe
#   salzburg
#   steiermark
#   kaernten
```

(continues on next page)

(continued from previous page)

```

# vorarlberg
# tirol
# religion
#authors =
# Erich Moechel

#[derstandard.at]
#ressorts =
# diskurs/kolumnen/rauscher
# inland/serienundblogs/standardabweichung
# etat
# immobilien
#users =
# 571924

#[arstechnica.com]
#channels =
# index
# features
# technology-lab
# gadgets
# business
# security
# tech-policy
# apple
# gaming
# science
# multiverse
# cars
# staff-blogs
# cardboard
# open-source
# microsoft
# software
# telecom
# web

#[momoxfashion.com]
#links =
# /de/herren?sortiertnach=neueste

#[kurier.at]
#channels =
# /chronik/wien
#articles =
# /meinung/pammesberger-2018-die-karikatur-zum-tag/309.629.015/slideshow
#authors =
# niki.glattauer
# guido.tartarotti
# florian.holzer
# barbara.kaufmann

#[spotify.com]
#market = AT
#shows =
# 6u7pI0o0CUBQq0T1fwPgbj

```

(continues on next page)

(continued from previous page)

```
#[wienerzeitung.at]
#ressorts =
#    nachrichten/politik/wien
#    nachrichten/politik
#    nachrichten/wirtschaft
#    meinung

#[ft.com]
#ressorts =
#    homepage
#    the-big-read

#[economist.com]
#ressorts =
#    finance-and-economics
#    special-report
#    leaders

#[tinyletter.com]
#accounts =
#    dabeaz

#[riskommunal]
#urls =
#    http://yourlocalcommunity.tld/News
#    https://mytown.tld/BUeRGERSERVICE/Neuigkeiten
```

Supported Websites

Feeds is currently able to create full text Atom feeds for the websites listed below. All feeds contain the articles in full text so you never have to leave your feed reader while reading.

4.1 A note on paywalls

Some sites (*Falter*, *Konsument*, *LWN*) offer articles only behind a paywall. If you have a paid subscription, you can configure your username and password in `feeds.cfg` (see also *Configure Feeds*) and also paywalled articles will be included in full text in the created feed. If you don't have a subscription and hence the full text cannot be included, paywalled articles are tagged with `paywalled` so they can be filtered, if desired.

4.2 Most popular sites

4.2.1 arstechnica.com

Full text feeds for *Ars Technica*.

Configuration

Add `arstechnica.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    arstechnica.com
```

`arstechnica.com` supports different channels via the `channels` parameter (one per line). If no channel is given, `features` is used. Go to *RSS feeds* for a list of all feeds.

```
[arstechnica.com]
channels =
    index
    features
    technology-lab
    gadgets
    business
    security
    tech-policy
    apple
    gaming
    science
    multiverse
    cars
    staff-blogs
    cardboard
    open-source
    microsoft
    software
    telecom
    web
```

4.2.2 economist.com

Newest articles from `economist.com`.

Configuration

Add `economist.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    economist.com
```

`economist.com` supports different ressorts via the `ressorts` parameter (one per line). See <https://www.economist.com/rss> for a list of ressorts.

Example configuration:

```
[economist.com]
ressorts =
    finance-and-economics
    special-report
    leaders
```

4.2.3 ft.com

Newest articles from `ft.com`.

Configuration

Add `ft.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    ft.com
```

ft.com supports different ressorts via the `ressorts` parameter (one per line). The ressort is the path in the URL (e. g. for <https://www.ft.com/companies/technology> the ressort is `companies/technology`). For the homepage the special ressort homepage can be used.

Example configuration:

```
[ft.com]
ressorts =
    homepage
    the-big-read
```

4.2.4 indiehackers.com

Newest interviews on Indie Hackers.

Configuration

Add `indiehackers.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    indiehackers.com
```

4.2.5 lwn.net

Newest articles from LWN with special treatment of LWN Weekly Editions. Please note that LWN requires the cache to be enabled to minimize useless requests. In case you provide username and password, the session (cookie) is also cached until the cache entry expires.

Configuration

Add `lwn.net` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    lwn.net
```

LWN has paywalled articles. If you want to crawl them, please provide username and password.

```
[lwn.net]
username =
password =
```

4.2.6 npr.org

Archive of the Planet Money Newsletter.

Configuration

Add `npr.org` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    npr.org
```

4.2.7 spotify.com

Podcasts hosted on Spotify.

Configuration

Add `spotify.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    spotify.com
```

The market you are in (i. e. your country as an ISO 3166-1 alpha-2 country code) has to be specified in the config as well. For example, for Austria specify: `market = AT`

spotify.com supports different podcasts via the `show` parameter (one per line).

Example configuration:

```
[spotify.com]
market = AT
shows =
    6u7pI0o0CUBQq0T1fwPgbj
```

4.2.8 vice.com

Newest articles from VICE.

Configuration

Add `vice.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    vice.com
```

VICE supports different locations via the `locales` parameter (one per line).

```
[vice.com]
locales =
    de_at
    de
```

4.3 Support for generic sites

4.3.1 Generic full-text extraction

The generic spider can transform already existing Atom or RSS feeds, which usually only contain a summary or a few lines of the content, into full content feeds. It is similar to [Full-Text RSS](#) but uses a port of an older version of [Readability](#) under the hood and currently doesn't support `site_config` files. It works best for blog articles.

Some feeds already provide the full content but in a tag that is not used by your feed reader. E.g. feeds created by Wordpress usually have the full content in the “encoded” tag. In such cases it's best to add the URL to the `fulltext_urls` entry which extracts the content directly from the feed without [Readability](#). There is a little helper script in [scripts/check-for-fulltext-content](#) to detect if a feed contains full-text content.

Configuration

Add `generic` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    generic
```

Add the feed URLs (Atom or XML) to the config file.

```
# List of URLs to RSS/Atom feeds to crawl, one per line.
[generic]
urls =
    https://www.example.com/feed.atom
    https://www.example.org/feed.xml
fulltext_urls =
    https://myblog.example.com/feed/
```

4.4 All supported sites

4.4.1 addendum.org

Newest articles from [Addendum](#).

Configuration

Add `addendum.org` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    addendum.org
```

4.4.2 ak.ciando.com

Most recently added books to the Arbeiterkammer e-library on [ak.ciando.com](#).

Configuration

Add `ak.ciando.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    ak.ciando.com
```

4.4.3 atv.at

Get newest episodes of TV shows from [ATV.at](#).

Configuration

Add `atv.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    atv.at
```

4.4.4 biblioweb.at

Most recently added media to libraries based on the [biblioweb.at](#) software.

Configuration

Add `biblioweb.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    biblioweb.at
```

The location of your library that uses [biblioweb.at](#) is needed as parameter.

```
[biblioweb.at]
location =
```

4.4.5 cbird.at

Newest releases of the [cbird](#) software.

Configuration

Add `cbird.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    cbird.at
```

4.4.6 derstandard.at

Newest articles from [derStandard.at](#).

Configuration

Add `derstandard.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    derstandard.at
```

`derstandard.at` supports different ressorts via the `ressorts` parameter (one per line).

The spider also has support user postings via the `users` parameter (one per line).

Example configuration:

```
[derstandard.at]
ressorts =
    diskurs/kolumnen/rauscher
    etat
    immobilien
users =
    4894
    571924
```

4.4.7 dietiwag.org

Latest articles of [dietiwag.org](#).

Configuration

Add `dietiwag.org` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    dietiwag.org
```

4.4.8 falter.at

Get newest articles and restaurant reviews from [Falter](#).

Configuration

Add `falter.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    falter.at
```

Falter has a paywall for certain articles. If you want to crawl paid articles, please provide `abonr` (subscription number) and `password`.

`pages` accepts `magazine` for the Falter newspaper and `lokal Fuehrer_reviews`, `lokal Fuehrer_newest` for restaurant and `streams` for movie streams. By default all are scraped.

`blogs` accepts `slugs` for the blogs from <https://cms.falter.at/blogs/>.

```
[falter.at]
abonr =
password =
pages =
    magazine
    lokal Fuehrer_reviews
    lokal Fuehrer_newest
    streams
blogs =
    lingers
    thinktank
```

4.4.9 keycloak.org

Blog posts from Keycloak.

Configuration

Add `keycloak.org` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    keycloak.org
```

4.4.10 konsument.at

Get newest articles from [konsument.at](https://www.konsument.at).

Configuration

Add `konsument.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    konsument.at
```

This website has a paywall for certain articles. If you want to crawl paid articles, please provide `username` and `password`:

```
[konsument.at]
username =
password =
```


4.4.11 kurier.at

Newest articles from [Kurier.at](#).

Configuration

Add `kurier.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    kurier.at
```

`kurier.at` supports different channels via the `channels` parameter, articles via the `articles` parameter and authors via the `authors` parameter (one per line).

Example configuration:

```
[kurier.at]
channels =
    /chronik/wien
articles =
    /meinung/pammesberger-2018-die-karikatur-zum-tag/309.629.015/slideshow
authors =
    niki.glattauer
    guido.tartarotti
    florian.holzer
    barbara.kaufmann
```

4.4.12 lbg.at

Latest articles from [LBG](#).

Configuration

Add `lbg.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    lb.g.at
```

4.4.13 momoxfashion.com

Items available for buying at [momoxfashion](#).

Configuration

Add `momoxfashion.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    momoxfashion.com
```

By default, [newest items](#) (from the first three pages) will be included. You can provide a list of links in case you want to limit the items to a specific brand or size.

```
[momoxfashion.com]
links =
    /katalog?sortiertnach=neueste
```

4.4.14 nachrichten.at

Newest articles from [Oberösterreichische Nachrichten](#).

Configuration

Add `nachrichten.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    nachrichten.at
```

[Oberösterreichische Nachrichten](#) supports different ressorts via the `ressorts` parameter (one per line). If no ressort is given, the default ressort “nachrichten” is used.

```
[nachrichten.at]
ressorts =
    linz
    wels
```

4.4.15 oe1.orf.at

Newest episodes of radio shows from [ORF Ö1](#).

Configuration

Add `oe1.orf.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    oe1.orf.at
```

4.4.16 orf.at

Newest articles from [ORF ON](#).

Configuration

Add `orf.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    orf.at
```

orf.at supports different channels via the `channels` parameter (one per line). If no channel is given, `news` is used. It is also possible to give a list of authors for which feeds will then be generated. Note that the channels in which the author writes still has to be included in the `channels` parameter.

```
[orf.at]
channels =
    burgenland
    fm4
    help
    kaernten
    news
    noe
    oe3
    oesterreich
    ooe
    religion
    salzburg
    science
    sport
    steiermark
    tirol
    vorarlberg
    wien
authors =
    Erich Moechel
```

4.4.17 profil.at

Newest articles from [profil](#).

Configuration

Add `profil.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    profil.at
```

4.4.18 puls4.com

Newest episodes of TV shows from [puls4.com](#).

Configuration

Add `puls4.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    puls4.com
```

4.4.19 python-patterns.guide

The latest articles from [python-patterns.guide](#). Since articles on [python-patterns.guide](#) do not have a publication date, the `Last-Modified` header is used for the updated field which might not be accurate or stable. I.e. old articles might have a newer value in the updated field even if they were not updated.

Configuration

Add `python-patterns.guide` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    python-patterns.guide
```

4.4.20 riskommunal

News from your local town or community, if their website is maintained with [RIS Kommunal](#).

Configuration

Add `riskommunal` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    riskommunal
```

At least one url is required. The local community or town website typically has a “News” or “Neuigkeiten” URL that you may use.

```
[riskommunal]
urls =
    http://yourlocalcommunity.tld/News
    https://mytown.tld/BUERGERSERVICE/Neuigkeiten
```

4.4.21 servustv.com

Videos shown on [ServusTV](#) in the next two weeks.

Configuration

Add `servustv.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    servustv.com
```

4.4.22 the oatmeal.com

Comics and blog posts from [The Oatmeal](#).

Configuration

Add `theoatmeal.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    theoatmeal.com
```

4.4.23 tinyletter.com

Latest articles from [tinyletter](#) users.

Configuration

Add `tinyletter.com` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    tinyletter.com
```

At least one account is required. The account name is visible on the subscription page, e.g. for <http://tinyletter.com/dabeaz>, the account name is `dabeaz`.

```
[tinyletter.com]
accounts =
    dabeaz
```

4.4.24 trend.at

Newest articles from [trend](#).

Configuration

Add `trend.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    trend.at
```

4.4.25 tuwien.ac.at

Newest Mitteilungsblätter issued by [TU Wien](#).

Configuration

Add `tuwien.ac.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    tuwien.ac.at
```

4.4.26 tvthek.orf.at

Newest episodes of TV shows from [ORF TVthek](#).

Configuration

Add `tvthek.orf.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    tvthek.orf.at
```

4.4.27 uebermedien.de

Newest articles from [Übermedien](#).

Configuration

Add `uebermedien.de` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    uebermedien.de
```

[Übermedien](#) has a paywall for certain articles. If you want to crawl paid articles, please provide your Blendle username and password.

```
[uebermedien.de]
username =
password =
```

4.4.28 usenix.org

Newest issues of the Usenix Magazine ;login:.

Configuration

Add `usenix.org` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    usenix.org
```

4.4.29 verbraucherrecht.at

Newest articles from [Verbraucherrecht](#).

Configuration

Add `verbraucherrecht.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    verbraucherrecht.at
```

4.4.30 wienerlinien.at

Get newest articles from [Wiener Linien](#).

Configuration

Add `wienerlinien.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    wienerlinien.at
```

4.4.31 wienerzeitung.at

Newest articles from [Wiener Zeitung](#).

Configuration

Add `wienerzeitung.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    wienerzeitung.at
```

`wienerzeitung.at` supports different ressorts via the `ressorts` parameter (one per line).

Example configuration:

```
[wienerzeitung.at]
ressorts =
    nachrichten/politik/wien
    nachrichten/politik
    nachrichten/wirtschaft
    meinung
```

4.4.32 zeit.diebin.at

Newest articles from [zeitdiebin](#).

Configuration

Add `zeit.diebin.at` to the list of spiders:

```
# List of spiders to run by default, one per line.
spiders =
    zeit.diebin.at
```

Supporting a new Website

Feeds already supports a number of websites (see *Supported Websites*) but adding support for a new website doesn't take too much time. All you need to do is write a so-called spider. A spider is a Python class that is used by Feeds to extract content from a website.

The feed generation pipeline looks like this:

1. A spider extracts the content (e.g. an article) that should be part of the feed from a website. The spider also tells Feeds how the content should be cleaned up, e.g. which HTML elements should be removed.
2. Feeds takes the content, cleans it up with the hints from the spider and some generic cleanup rules (e.g. `<script>` tags are always removed).
3. Feeds writes an Atom feed for that site with the cleaned content to the file system.

5.1 A quick example

Writing a spider is easy! For simple websites it can be done in only about 30 lines of code.

Consider this example for a fictional website that hosts articles. When a new article is published, a link to it is added to an overview page. The idea is now to use that URL as a starting point for the spider and let the spider extract all the URLs to the articles. In the next step, the spider visits every article, extracts the article text and meta information (time, author) and creates a feed item out of it.

The following code shows how such a spider could look like for our example website:

```
import scrapy

from feeds.loaders import FeedEntryItemLoader
from feeds.spiders import FeedsSpider

class ExampleComSpider(FeedsSpider):
    name = "example.com"
    start_urls = ["https://www.example.com/articles"]
```

(continues on next page)

(continued from previous page)

```

feed_title = "Example Website"

def parse(self, response):
    article_links = response.css(".article__link::attr(href)").extract()
    for link in article_links:
        yield scrapy.Request(response.urljoin(link), self._parse_article)

def _parse_article(self, response):
    remove_elems = [".shareable-quote", ".share-bar"]
    il = FeedEntryItemLoader(
        response=response,
        base_url="https://{}/".format(self.name),
        remove_elems=remove_elems,
    )
    il.add_value("link", response.url)
    il.add_css("title", "h1::text")
    il.add_css("author_name", "header .user-link__name::text")
    il.add_css("content_html", ".article-body")
    il.add_css("updated", ".article-date::text")
    return il.load_item()

```

First, the URL from the `start_urls` list is downloaded and the response is given to `parse()`. From there we extract the article links that should be scraped and yield `scrapy.Request` objects from the for loop. The callback method `_parse_article()` is executed once the download has finished. It extracts the article from the response HTML document and returns an item that will be placed into the feed automatically.

It's enough to place the spider in the `spiders` folder. It doesn't have to be registered somewhere for Feeds to pick it up. Now you can run it:

```
$ feeds crawl example.com
```

The resulting feed can be found in `output/example.com/feed.xml`.

5.2 Reusing an existing feed

Often websites provide a feed but it's not full text. In such cases you usually only want to augment the original feed with the full article.

5.2.1 Generic spider

For a lot of feeds (especially those from blogs) it is actually sufficient to use the *Generic full-text extraction* spider which can extract content from any website using heuristics (go to *Generic full-text extraction* for more on that).

Note that a lot of feeds (e.g. those generated by Wordpress) actually contain the full text but your feed reader chooses to show a summary instead. In such cases you can also use the *Generic full-text extraction* spider and add your feed URL to the `fulltext_urls` key in the config. This will create a full text feed from an existing feed without having to rely on heuristics.

5.2.2 Custom extraction

These spiders take an existing RSS feed and inline the article content while cleaning up the content (removing share buttons, etc.):

- *arstechnica.com*
- *derstandard.at*
- *dietiwag.org*
- *ft.com*
- *lwn.net*
- *orf.at*
- *profil.at*

5.2.3 Paywalled content

If your website has a feed but some or all articles are behind a paywall or require to login to read, take a look at the following spiders:

- *lwn.net*
- *nachrichten.at*
- *uebermedien.de*

5.3 Creating a feed from scratch

Some websites don't offer any feed at all. In such cases we have to find an efficient way to detect new content and extract it.

5.3.1 Utilizing an API

Some use a REST API which we can use to fetch the content.

- *addendum.org*
- *falter.at*
- *indiehackers.com*
- *kurier.at*
- *oe1.orf.at*
- *tvthek.orf.at*
- *vice.com*

5.3.2 Utilizing the sitemap

Others provide a `sitemap` which we can parse:

- *trend.at*

5.3.3 Custom extraction

The last resort is to find a page that lists the newest articles and start scraping from there.

- *ak.ciendo.com*
- *atv.at*
- *biblioweb.at*
- *cbird.at*
- *economist.com*
- *lbg.at*
- *npr.org*
- *puls4.com*
- *python-patterns.guide*
- *riskommunal*
- *servustv.com*
- *tinyletter.com*
- *tuwien.ac.at*
- *momoxfashion.com*
- *usenix.org*
- *verbraucherrecht.at*
- *wienerlinien.at*
- *zeit.diebin.at*

For paywalled content, take a look at:

- *falter.at*
- *konsument.at*

5.4 Extraction rules

A great feed transports all the information from the original site but without the clutter. The reader should never have to leave their reader and go to the original site. The following rules help to reach that goal.

5.4.1 Unwanted content

Advertisement, share buttons/links, navigation elements and everything that is not part of the content is removed. The output should be similar to what Firefox Reader View (Readability) outputs, but more polished.

5.4.2 Images

The HTML tags `<figure>` and `<figcaption>` are used for figures (if possible). Example:

```
<figure>
<div></img><div>
<figcaption>A very interesting image.</figcaption>
</figure>
```

Credits for images are removed. Images are included in their highest resolution available.

5.4.3 Depaginate

If content is split in multiple pages, all pages are scraped.

5.4.4 Iframes

Iframes are removed if they are unnecessary or untouched. Iframes are automatically replaced with a link to their source.

5.4.5 Updated field

Every feed item has an updated field. If the spider cannot provide such a field for an item because the original site doesn't expose that information, Feeds will automatically use the timestamp when it saw the link of the item for the first time.

5.4.6 Not embeddable content

Sometimes external content like videos cannot be included in the feed because it needs JavaScript. In such cases the container of the external video is replaced with a note that says that the content is only available in the original content.

5.4.7 Regular expressions

Regular expressions are only used to replace content if using CSS selectors with `replace_elems` is not possible.

5.4.8 Categories

A feed item has categories taken from its original feed or from the site.

5.4.9 Headings

`<h*>` tags are used for headings (i. e. not generic tags like `<p>` or `<div>`). Headings start with `<h2>`. The title of the content is not part of the content and is removed.

5.4.10 Author name(s)

The name of all authors are added to the `author_name` field. The names are not part of the content and are removed.

CHAPTER 6

Docker

If you prefer to run Feeds in a docker container, you can use the official [PyFeeds image](#).

A `docker-compose.yml` could look like this:

```
version: "3.9"
services:
  pyfeeds:
    image: pyfeeds/pyfeeds:latest
    volumes:
      - ./config:/config
      - pyfeeds-output:/output
    command: --config /config/feeds.cfg crawl
volumes:
  pyfeeds-output:
    name: pyfeeds-output
```

It mounts the `config` folder next to the `docker-compose.yml` and uses the contained `feeds.cfg` as config for Feeds. The feeds are stored in a volume which could be picked up by a webserver:

```
version: "3.9"
services:
  pyfeeds-server:
    image: nginx:stable-alpine
    restart: always
    volumes:
      - pyfeeds-output:/usr/share/nginx/html:ro
volumes:
  pyfeeds-output:
    external: true
    name: pyfeeds-output
```

Now any other container in the same docker network (f.e. a ttrss server) could access the feeds (f.e. <http://pyfeeds-server/theoatmeal.com/feed.atom>). Add a port mapping in case you want to allow access from outside the container's docker network.

If you want to support a custom website, take a look at *Supporting a new Website*.

7.1 Spider class

A spider is a class in a module (Python file) in `feeds.spiders` that is a subclass of `feeds.spiders.FeedsSpider`, `feeds.spiders.FeedsCrawlSpider` or `feeds.spiders.FeedsXMLFeedSpider`.

- `FeedsXMLFeedSpider` is used, if the spider is based on parsing an XML document as a basis. This is useful if the spider should start from an existing XML feed or a sitemap.
- `FeedsCrawlSpider` is used, if the spider should crawl the site based on links that are found on the site. Patterns can be given to limit what links should be followed.
- `FeedsSpider` is used in all other cases (this spider is usually used).

7.1.1 Class variables

- `name`: The name of the spider (**mandatory**).
- `start_urls`: A list of URLs to start (used if the `start_requests(self)` method is not overwritten).
- `feed_title`: Title of the feed.
- `feed_subtitle`: Subtitle of the feed.
- `feed_link`
- `author_name`: Author of the feed.
- `feed_icon`: URL of a site favicon.
- `feed_logo`: URL of a site logo.

7.1.2 Methods

- `start_requests(self)`: If the start request is more complicated than a simply GET to the URL(s) in the `start_urls` list, this method can be overwritten. It is expected to yield or return a `scrapy.Request` object. Please note that this method can *only* emit `Request` objects.
- `parse(self, response)`: After a URL from `start_urls` has been scraped, the `parse()` method is called and the response is given as an argument. It is also the default call back method for new `scrapy.Request` objects.
- `parse_node(self, response, node)`: A `FeedsXMLFeedSpider` calls `parse_node()` instead of `parse()` for every node in the XML document returned by the URL in `start_urls`.

7.2 FeedEntryItemLoader

A spider uses a `FeedEntryItemLoader` object to extract content from a response. The following fields are accepted and can be added to a item loader object:

- `link`
- `title`
- `author_name`
- `author_email`
- `content_html`
- `updated`
- `category`
- `path`
- `enclosure_iri`
- `enclosure_type`

A value can be added to an item loader with the `add_value()`, `add_css()` or `add_xpath()` methods like in the following example:

```
il = FeedEntryItemLoader(response=response)
il.add_value("link", response.url)
il.add_css("title", "h1::text")
il.add_css("author_name", "header .user-link__name::text")
il.add_css("content_html", ".interview-body")
il.add_css("updated", ".date::text")
return il.load_item()
```

Only the `link` field is required, all the other fields can be empty but usually it is advised to add as many fields as possible (i.e. the original site provides).

If the `updated` field is not provided, the date and time during the extraction is used. If caching is enabled, the date and time when the item was first seen is cached and reused on following runs.

7.3 Input processing

Automatic rules are applied to fields depending on their type.

7.3.1 Default input rules

These rules are usually applied to every field.

1. Empty strings and `None` are skipped.
2. The content is stripped.
3. The content is unescaped twice, i.e. `& &xxx;` is converted to its decoded (binary) equivalent.

7.3.2 title

1. The default input rules apply.
2. One title: “<title 1>”
3. Two titles: “<title 1>: <title 2>”
4. Three or more titles: “<title 1>: <title 2> - <title 3> - <title n>”

7.3.3 updated

1. Empty strings and `None` are skipped.
2. Unless the date is already a `datetime` object, it is parsed using `dateutil.parser.parse()` (with the year expected to be first, and the day *not* expected to be first). If `dateutil` can't parse it because it's a human readable string, `dateparser` is used. `dayfirst` (default `False`), `yearfirst` (default `True`) and `ignoretz` (default `False`) can be set in the `FeedEntryItemLoader`.
3. If the `datetime` object is not already `timezone` aware, the `timezone` specified in the `FeedEntryItemLoader` is set.
4. The first `datetime` object is used.

7.3.4 author_name

1. The default input rules apply.
2. Multiple author names are joined with “, ” (comma and space) as a separator.

7.3.5 path

1. The default input rules apply.
2. Multiple paths are joined with `os.sep` (e.g. `/`) as a separator.

7.3.6 content_html

1. Empty strings and `None` are skipped.
2. `replace_regex` in the `FeedEntryItemLoader` is a dict with `pattern` as a key and `repl` as a value. `pattern` and `repl` are used as parameters for `re.sub()`. `pattern` can be a string or a pattern object, `repl` a string or a function.

3. `convert_footnotes` in the `FeedEntryItemLoader` is a list of CSS selectors which select footnotes or otherwise hidden text. Such elements are replaced with `<small>` elements and the text of the respective footnote in brackets.
4. `pullup_elems` in the `FeedEntryItemLoader` is a dict with a CSS selector as a key and a distance as a value. A parent that is a given distance away from the selected element is replaced with the selected element. E.g. a distance of 1 means that the children replaces its parent.
5. `replace_elems` in the `FeedEntryItemLoader` is a dict that contains a selector as a key and a string as a value. The selected element is replaced with the HTML fragment.
6. `remove_elems` in the `FeedEntryItemLoader` is a list with CSS selectors of elements that should be removed.
7. `remove_elems_xpath` in the `FeedEntryItemLoader` is a list with XPath queries of elements that should be removed.
8. `change_attribs` in the `FeedEntryItemLoader` is a dict with a CSS selector as a key and a dict that describes how to change attribs as a value. The dict contains the old attrib name as a key and the new attrib name as a value. If the value is `None`, the attrib is removed.
9. `change_tags` in the `FeedEntryItemLoader` is a dict with a CSS selector as a key and a new tag name as a value. The tag name of the selected element is changed to the new tag name.
10. Attributes `class`, `id` and ones that start with `data-` are removed.
11. Iframes are converted to a `<div>` that contains a link to the source of the iframe.
12. Scripts, JavaScript, comments, styles and inline styles are removed.
13. The HTML tree is flattened: Elements which do not have a text and are not supposed to be empty are removed. An element is replaced with its child if it has exactly one child and the child has the same tag.
14. References in tags like `<a>` and `` are made absolute.

Feeds uses [GitHub](#) as development platform.

8.1 Issues

- Search the existing issues in the [issue tracker](#).
- File a [new issue](#) in case the issue is undocumented.

8.2 Pull requests

- Fork the project to your private repository.
- Create a topic branch and make your desired changes.
- Open a pull request. Make sure the GitHub CI checks are passing.

CHAPTER 9

License

GNU AFFERO GENERAL PUBLIC LICENSE
Version 3, 19 November 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone **is** permitted to copy **and** distribute verbatim copies
of this license document, but changing it **is not** allowed.

Preamble

The GNU Affero General Public License **is** a free, copyleft license **for**
software **and** other kinds of works, specifically designed to ensure
cooperation **with** the community **in** the case of network server software.

The licenses **for** most software **and** other practical works are designed
to take away your freedom to share **and** change the works. By contrast,
our General Public Licenses are intended to guarantee your freedom to
share **and** change **all** versions of a program--to make sure it remains free
software **for** **all** its users.

When we speak of free software, we are referring to freedom, **not**
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (**and** charge **for**
them **if** you wish), that you receive source code **or** can get it **if** you
want it, that you can change the software **or** use pieces of it **in** new
free programs, **and** that you know you can do these things.

Developers that use our General Public Licenses protect your rights
with two steps: (1) **assert** copyright on the software, **and** (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

A secondary benefit of defending **all** users' freedom **is** that
improvements made **in** alternate versions of the program, **if** they
receive widespread use, become available **for** other developers to

(continues on next page)

(continued from previous page)

incorporate. Many developers of free software are heartened **and** encouraged by the resulting cooperation. However, **in** the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version **and** letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License **is** designed specifically to ensure that, **in** such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License **and** published by Affero, was designed to accomplish similar goals. This **is** a different license, **not** a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms **and** conditions **for** copying, distribution **and** modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such **as** semiconductor masks.

"The Program" refers to **any** copyrightable work licensed under this License. Each licensee **is** addressed **as** "you". "Licensees" **and** "recipients" may be individuals **or** organizations.

To "modify" a work means to copy **from** **or** adapt **all** **or** part of the work **in** a fashion requiring copyright permission, other than the making of an exact copy. The resulting work **is** called a "modified version" of the earlier work **or** a work "based on" the earlier work.

A "covered work" means either the unmodified Program **or** a work based on the Program.

To "propagate" a work means to do anything **with** it that, without permission, would make you directly **or** secondarily liable **for** infringement under applicable copyright law, **except** executing it on a computer **or** modifying a private copy. Propagation includes copying, distribution (**with** **or** without modification), making available to the public, **and** **in** some countries other activities **as** well.

To "convey" a work means **any** kind of propagation that enables other parties to make **or** receive copies. Mere interaction **with** a user through a computer network, **with** no transfer of a copy, **is not** conveying.

An interactive user interface displays "Appropriate Legal Notices"

(continues on next page)

(continued from previous page)

to the extent that it includes a convenient **and** prominently visible feature that (1) displays an appropriate copyright notice, **and** (2) tells the user that there **is** no warranty **for** the work (**except** to the extent that warranties are provided), that licensees may convey the work under this License, **and** how to view a copy of this License. If the interface presents a **list** of user commands **or** options, such **as** a menu, a prominent item **in** the **list** meets this criterion.

1. Source Code.

The **"source code"** **for** a work means the preferred form of the work **for** making modifications to it. **"Object code"** means **any** non-source form of a work.

A **"Standard Interface"** means an interface that either **is** an official standard defined by a recognized standards body, **or**, **in** the case of interfaces specified **for** a particular programming language, one that **is** widely used among developers working **in** that language.

The **"System Libraries"** of an executable work include anything, other than the work **as** a whole, that (a) **is** included **in** the normal form of packaging a Major Component, but which **is not** part of that Major Component, **and** (b) serves only to enable use of the work **with** that Major Component, **or** to implement a Standard Interface **for** which an implementation **is** available to the public **in** source code form. A **"Major Component"**, **in** this context, means a major essential component (kernel, window system, **and** so on) of the specific operating system (**if any**) on which the executable work runs, **or** a compiler used to produce the work, **or** an **object** code interpreter used to run it.

The **"Corresponding Source"** **for** a work **in** **object** code form means **all** the source code needed to generate, install, **and** (**for** an executable work) run the **object** code **and** to modify the work, including scripts to control those activities. However, it does **not** include the work's System Libraries, **or** general-purpose tools **or** generally available free programs which are used unmodified **in** performing those activities but which are **not** part of the work. For example, Corresponding Source includes interface definition files associated **with** source files **for** the work, **and** the source code **for** shared libraries **and** dynamically linked subprograms that the work **is** specifically designed to require, such **as** by intimate data communication **or** control flow between those subprograms **and** other parts of the work.

The Corresponding Source need **not** include anything that users can regenerate automatically **from** **other** parts of the Corresponding Source.

The Corresponding Source **for** a work **in** source code form **is** that same work.

2. Basic Permissions.

All rights granted under this License are granted **for** the term of copyright on the Program, **and** are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output **from** **running** a covered work **is** covered by this License only **if** the output, given its

(continues on next page)

(continued from previous page)

content, constitutes a covered work. This License acknowledges your rights of fair use **or** other equivalent, **as** provided by copyright law.

You may make, run **and** propagate covered works that you do **not** convey, without conditions so long **as** your license otherwise remains **in** force. You may convey covered works to others **for** the sole purpose of having them make modifications exclusively **for** you, **or** provide you **with** facilities **for** running those works, provided that you comply **with** the terms of this License **in** conveying **all** material **for** which you do **not** control copyright. Those thus making **or** running the covered works **for** you must do so exclusively on your behalf, under your direction **and** control, on terms that prohibit them **from making** any copies of your copyrighted material outside their relationship **with** you.

Conveying under **any** other circumstances **is** permitted solely under the conditions stated below. Sublicensing **is not** allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under **any** applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, **or** similar laws prohibiting **or** restricting circumvention of such measures.

When you convey a covered work, you waive **any** legal power to forbid circumvention of technological measures to the extent such circumvention **is** effected by exercising rights under this License **with** respect to the covered work, **and** you disclaim **any** intention to limit operation **or** modification of the work **as** a means of enforcing, against the work's users, your **or** third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's **source code** as you receive it, **in any** medium, provided that you conspicuously **and** appropriately publish on each copy an appropriate copyright notice; keep intact **all** notices stating that this License **and any** non-permissive terms added **in** accord **with** section 7 apply to the code; keep intact **all** notices of the absence of **any** warranty; **and** give **all** recipients a copy of this License along **with** the Program.

You may charge **any** price **or** no price **for** each copy that you convey, **and** you may offer support **or** warranty protection **for** a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, **or** the modifications to produce it **from the** Program, **in** the form of source code under the terms of section 4, provided that you also meet **all** of these conditions:

- a) The work must carry prominent notices stating that you modified it, **and** giving a relevant date.
- b) The work must carry prominent notices stating that it **is**

(continues on next page)

(continued from previous page)

released under this License **and any** conditions added under section 7. This requirement modifies the requirement **in** section 4 to "keep intact all notices".

c) You must license the entire work, **as** a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along **with any** applicable section 7 additional terms, to the whole of the work, **and all** its parts, regardless of how they are packaged. This License gives no permission to license the work **in any** other way, but it does **not** invalidate such permission **if** you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, **if** the Program has interactive interfaces that do **not** display Appropriate Legal Notices, your work need **not** make them do so.

A compilation of a covered work **with** other separate **and** independent works, which are **not** by their nature extensions of the covered work, **and** which are **not** combined **with** it such **as** to form a larger program, **in or** on a volume of a storage **or** distribution medium, **is** called an "aggregate" **if** the compilation **and** its resulting copyright are **not** used to limit the access **or** legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work **in** an aggregate does **not** cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work **in object** code form under the terms of sections 4 **and** 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, **in** one of these ways:

a) Convey the **object** code **in, or** embodied **in**, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used **for** software interchange.

b) Convey the **object** code **in, or** embodied **in**, a physical product (including a physical distribution medium), accompanied by a written offer, valid **for** at least three years **and** valid **for as** long **as** you offer spare parts **or** customer support **for** that product model, to give anyone who possesses the **object** code either (1) a copy of the Corresponding Source **for all** the software **in** the product that **is** covered by this License, on a durable physical medium customarily used **for** software interchange, **for** a price no more than your reasonable cost of physically performing this conveying of source, **or** (2) access to copy the Corresponding Source **from a** network server at no charge.

c) Convey individual copies of the **object** code **with** a copy of the written offer to provide the Corresponding Source. This alternative **is** allowed only occasionally **and** noncommercially, **and** only **if** you received the **object** code **with** such an offer, **in** accord **with** subsection 6b.

(continues on next page)

(continued from previous page)

d) Convey the `object` code by offering access `from a` designated place (gratis `or for` a charge), `and` offer equivalent access to the Corresponding Source `in` the same way through the same place at no further charge. You need `not` require recipients to copy the Corresponding Source along `with` the `object` code. If the place to copy the `object` code `is` a network server, the Corresponding Source may be on a different server (operated by you `or` a third party) that supports equivalent copying facilities, provided you maintain clear directions `next` to the `object` code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it `is` available `for as long as` needed to satisfy these requirements.

e) Convey the `object` code using peer-to-peer transmission, provided you inform other peers where the `object` code `and` Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the `object` code, whose source code `is` excluded `from the` Corresponding Source `as` a System Library, need `not` be included `in` conveying the `object` code work.

A "User Product" `is` either (1) a "consumer product", which means any tangible personal `property` which `is` normally used `for` personal, family, `or` household purposes, `or` (2) anything designed `or` sold `for` incorporation into a dwelling. In determining whether a product `is` a consumer product, doubtful cases shall be resolved `in` favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical `or` common use of that `class of` product, regardless of the status of the particular user `or` of the way `in` which the particular user actually uses, `or` expects `or is` expected to use, the product. A product `is` a consumer product regardless of whether the product has substantial commercial, industrial `or` non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" `for` a User Product means any methods, procedures, authorization keys, `or` other information required to install `and` execute modified versions of a covered work `in` that User Product `from a` modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified `object` code `is in` no case prevented `or` interfered `with` solely because modification has been made.

If you convey an `object` code work under this section `in, or with, or` specifically `for` use `in`, a User Product, `and` the conveying occurs `as` part of a transaction `in` which the right of possession `and` use of the User Product `is` transferred to the recipient `in` perpetuity `or for` a fixed term (regardless of how the transaction `is` characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does `not` apply `if` neither you nor any third party retains the ability to install modified `object` code on the User Product (`for` example, the work has been installed `in` ROM).

The requirement to provide Installation Information does `not` include a requirement to `continue` to provide support service, warranty, `or` updates `for` a work that has been modified `or` installed by the recipient, `or for`

(continues on next page)

(continued from previous page)

the User Product **in** which it has been modified **or** installed. Access to a network may be denied when the modification itself materially **and** adversely affects the operation of the network **or** violates the rules **and** protocols **for** communication across the network.

Corresponding Source conveyed, **and** Installation Information provided, **in** accord **with** this section must be **in** a **format** that **is** publicly documented (**and with** an implementation available to the public **in** source code form), **and** must require no special password **or** key **for** unpacking, reading **or** copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions **from one or** more of its conditions. Additional permissions that are applicable to the entire Program shall be treated **as** though they were included **in** this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove **any** additional permissions **from that** copy, **or from any** part of it. (Additional permissions may be written to require their own removal **in** certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, **for** which you have **or** can give appropriate copyright permission.

Notwithstanding **any** other provision of this License, **for** material you add to a covered work, you may (**if** authorized by the copyright holders of that material) supplement the terms of this License **with** terms:

- a) Disclaiming warranty **or** limiting liability differently **from the** terms of sections 15 **and** 16 of this License; **or**
- b) Requiring preservation of specified reasonable legal notices **or** author attributions **in** that material **or in** the Appropriate Legal Notices displayed by works containing it; **or**
- c) Prohibiting misrepresentation of the origin of that material, **or** requiring that modified versions of such material be marked **in** reasonable ways **as** different **from the** original version; **or**
- d) Limiting the use **for** publicity purposes of names of licensors **or** authors of the material; **or**
- e) Declining to grant rights under trademark law **for** use of some trade names, trademarks, **or** service marks; **or**
- f) Requiring indemnification of licensors **and** authors of that material by anyone who conveys the material (**or** modified versions of it) **with** contractual assumptions of liability to the recipient, **for** **any** liability that these contractual assumptions directly impose on those licensors **and** authors.

All other non-permissive additional terms are considered "**further**

(continues on next page)

(continued from previous page)

restrictions" within the meaning of section 10. If the Program as you received it, **or** any part of it, contains a notice stating that it **is** governed by this License along **with** a term that **is** a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing **or** conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does **not** survive such relicensing **or** conveying.

If you add terms to a covered work **in** accord **with** this section, you must place, **in** the relevant source files, a statement of the additional terms that apply to those files, **or** a notice indicating where to find the applicable terms.

Additional terms, permissive **or** non-permissive, may be stated **in** the form of a separately written license, **or** stated **as** exceptions; the above requirements apply either way.

8. Termination.

You may **not** propagate **or** modify a covered work **except as** expressly provided under this License. Any attempt otherwise to propagate **or** modify it **is** void, **and** will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, **if** you cease all violation of this License, then your license **from a** particular copyright holder **is** reinstated (a) provisionally, unless **and** until the copyright holder explicitly **and finally** terminates your license, **and** (b) permanently, **if** the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license **from a** particular copyright holder **is** reinstated permanently **if** the copyright holder notifies you of the violation by some reasonable means, this **is** the first time you have received notice of violation of this License (**for any work**) **from that** copyright holder, **and** you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does **not** terminate the licenses of parties who have received copies **or** rights **from you** under this License. If your rights have been terminated **and not** permanently reinstated, you do **not** qualify to receive new licenses **for** the same material under section 10.

9. Acceptance Not Required **for** Having Copies.

You are **not** required to accept this License **in** order to receive **or** run a copy of the Program. Ancillary propagation of a covered work occurring solely **as** a consequence of using peer-to-peer transmission to receive a copy likewise does **not** require acceptance. However, nothing other than this License grants you permission to propagate **or** modify any covered work. These actions infringe copyright **if** you do **not** accept this License. Therefore, by modifying **or** propagating a covered work, you indicate your acceptance of this License to do so.

(continues on next page)

(continued from previous page)

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license **from the** original licensors, to run, modify **and** propagate that work, subject to this License. You are **not** responsible **for** enforcing compliance by third parties **with** this License.

An "entity transaction" **is** a transaction transferring control of an organization, **or** substantially **all** assets of one, **or** subdividing an organization, **or** merging organizations. If propagation of a covered work results **from an** entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work **from the** predecessor **in** interest, **if** the predecessor has it **or** can get it **with** reasonable efforts.

You may **not** impose **any** further restrictions on the exercise of the rights granted **or** affirmed under this License. For example, you may **not** impose a license fee, royalty, **or** other charge **for** exercise of rights granted under this License, **and** you may **not** initiate litigation (including a cross-claim **or** counterclaim **in** a lawsuit) alleging that **any** patent claim **is** infringed by making, using, selling, offering **for** sale, **or** importing the Program **or** **any** portion of it.

11. Patents.

A "contributor" **is** a copyright holder who authorizes use under this License of the Program **or** a work on which the Program **is** based. The work thus licensed **is** called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned **or** controlled by the contributor, whether already acquired **or** hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, **or** selling its contributor version, but do **not** include claims that would be infringed only **as** a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses **in** a manner consistent **with** the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer **for** sale, **import and** otherwise run, modify **and** propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" **is** **any** express agreement **or** commitment, however denominated, **not** to enforce a patent (such **as** an express permission to practice a patent **or** covenant **not** to sue **for** patent infringement). To "grant" such a patent license to a party means to make such an agreement **or** commitment **not** to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, **and** the Corresponding Source of the work **is not** available **for** anyone to copy, free of charge **and** under the terms of this License, through a publicly available network server **or** other readily accessible means,

(continues on next page)

(continued from previous page)

then you must either (1) cause the Corresponding Source to be so available, **or** (2) arrange to deprive yourself of the benefit of the patent license **for** this particular work, **or** (3) arrange, **in** a manner consistent **with** the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but **for** the patent license, your conveying the covered work **in** a country, **or** your recipient's use of the covered work **in** a country, would infringe one **or** more identifiable patents **in** that country that you have reason to believe are valid.

If, pursuant to **or in** connection **with** a single transaction **or** arrangement, you convey, **or** propagate by procuring conveyance of, a covered work, **and** grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify **or** convey a specific copy of the covered work, then the patent license you grant **is** automatically extended to **all** recipients of the covered work **and** works based on it.

A patent license **is** "discriminatory" **if** it does **not** include within the scope of its coverage, prohibits the exercise of, **or is** conditioned on the non-exercise of one **or** more of the rights that are specifically granted under this License. You may **not** convey a covered work **if** you are a party to an arrangement **with** a third party that **is in** the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, **and** under which the third party grants, to **any** of the parties who would receive the covered work **from you**, a discriminatory patent license (a) **in** connection **with** copies of the covered work conveyed by you (**or** copies made **from those** copies), **or** (b) primarily **for and in** connection **with** specific products **or** compilations that contain the covered work, unless you entered into that arrangement, **or** that patent license was granted, prior to 28 March 2007.

Nothing **in** this License shall be construed **as** excluding **or** limiting **any** implied license **or** other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement **or** otherwise) that contradict the conditions of this License, they do **not** excuse you **from the** conditions of this License. If you cannot convey a covered work so **as** to satisfy simultaneously your obligations under this License **and any** other pertinent obligations, then **as** a consequence you may **not** convey it at **all**. For example, **if** you agree to terms that obligate you to collect a royalty **for** further conveying **from those** to whom you convey the Program, the only way you could satisfy both those terms **and** this License would be to refrain entirely **from conveying** the Program.

13. Remote Network Interaction; Use **with** the GNU General Public License.

Notwithstanding **any** other provision of this License, **if** you modify the Program, your modified version must prominently offer **all** users interacting **with** it remotely through a computer network (**if** your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source **from a** network server at no charge, through some standard **or** customary

(continues on next page)

(continued from previous page)

means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source **for any** work covered by version 3 of the GNU General Public License that **is** incorporated pursuant to the following paragraph.

Notwithstanding **any** other provision of this License, you have permission to link **or** combine **any** covered work **with** a work licensed under version 3 of the GNU General Public License into a single combined work, **and** to convey the resulting work. The terms of this License will **continue** to apply to the part which **is** the covered work, but the work **with** which it **is** combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

The Free Software Foundation may publish revised **and/or** new versions of the GNU Affero General Public License **from time** to time. Such new versions will be similar **in** spirit to the present version, but may differ **in** detail to address new problems **or** concerns.

Each version **is** given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "**or any later version**" applies to it, you have the option of following the terms **and** conditions either of that numbered version **or** of **any** later version published by the Free Software Foundation. If the Program does **not** specify a version number of the GNU Affero General Public License, you may choose **any** version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version **for** the Program.

Later license versions may give you additional **or** different permissions. However, no additional obligations are imposed on **any** author **or** copyright holder **as** a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "**AS IS**" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

(continues on next page)

(continued from previous page)

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see <http://www.gnu.org/licenses/>.

CHAPTER 10

About Feeds

Once upon a time every website offered an RSS feed to keep readers updated about new articles/blog posts via the users' feed readers. These times are long gone. The once iconic orange RSS icon has been replaced by "social share" buttons.

Feeds aims to bring back the good old reading times. It creates Atom feeds for websites that don't offer them (any-more). It allows you to read new articles of your favorite websites in your feed reader (e.g. [TinyTinyRSS](#)) even if this is not officially supported by the website.

Furthermore it can also enhance existing feeds by inlining the actual content into the feed entry so it can be read without leaving the feed reader.

Feeds is based on [Scrapy](#), a framework for extracting data from websites and it has support for a few websites already, see [Supported Websites](#). It's easy to add support for new websites. Just take a look at the existing [spiders](#) and feel free to open a [pull request](#)!

CHAPTER 11

Related work

- [morss](#) creates feeds, similar to Feeds but in “real-time”, i.e. on (HTTP) request.
- [Full-Text RSS](#) converts feeds to contain the full article and not only a teaser based on heuristics and rules. Feeds are converted in “real-time”, i.e. on request basis.
- [f43.me](#) converts feeds to contain the full article and also improves articles by adding links to the comment sections of Hacker News and Reddit. Feeds are converted periodically.
- [python-fts](#) is a library to extract content from pages. A partial reimplementation of Full-Text RSS.

CHAPTER 12

Authors

Feeds is written and maintained by [Florian Preinstorfer](#) and [Lukas Anzinger](#).