
pypo Documentation

Release 0.1

Jens Kadenbach

January 19, 2015

1	Configuration	1
2	SECRET_KEY	3
3	DEBUG, TEMPLATE_DEBUG, CRISPY_FAIL_SILENTLY	5
4	ALLOWED_HOSTS	7
5	ADMINS	9
6	STATIC_ROOT	11
7	STATIC_URL	13
8	DATABASES	15
9	HAYSTACK_CONNECTIONS	17
10	Readme Module	19
10.1	Models	19
10.2	Scrapers	20
11	About Pypo	21
11.1	Documentation	21
11.2	Features	21
11.3	Installation	22
11.4	Deploying	22
11.5	Requirements status	22
11.6	License	22
12	Indices and tables	23
	Python Module Index	25

Configuration

You can overwrite all default settings either directly in `pypo/settings.py` or you create the file `pypo/settings_local.py` which is imported in the `pypo/settings.py` and therefore can overwrite all settings.

SECRET_KEY

Like in all django application, you have to set a unique secret key. [Django SECRET_KEY documentation](#)

DEBUG, TEMPLATE_DEBUG, CRISPY_FAIL_SILENTLY

To enable or disable debugging (crispy is a form component)

ALLOWED_HOSTS

A list of hostnames. [Django ALLOWED_HOSTS documentation](#)

ADMINS

A list of tuples (“name”, “email”) of admins

STATIC_ROOT

Absolute path where your static file are collected to when you call `./manage.py collectstatic`

STATIC_URL

Url where those files are available

DATABASES

You database config. Pypo is tested with PostgreSQL, but any django supported DB should be fine. [Django DATABASES documentation](#)

HAYSTACK_CONNECTIONS

If you want to use something else than Whoosh (a pure python search index), you can configure the search backend here. [Django Haystack documentation](#) It is recommended to switch to Elasticsearch for larger datasets:

```
HAYSTACK_CONNECTIONS = {  
    'default': { 'ENGINE': 'haystack.backends.elasticsearch_backend.ElasticsearchSearchEngine',  
                 'URL': 'http://127.0.0.1:9200/', 'INDEX_NAME': 'pypo',  
    },  
}
```

Readme Module

Everything except for configuration.

10.1 Models

```
class readme.models.Item(*args, **kwargs)
    Entry in the read-it-later-list

    created = None
        param created Creating date of the item

    fetch_article()
        Fetches a title and a readable_article for the current url. It uses the scrapers module for this and only
        downloads the content.

    owner
        param owner Owning user

    readable_article = None
        param readable_article Processed content of the url

    safe_article = None
        param safe_article Escaped and stripped of tags

    tags = <taggit.managers._TaggableManager object at 0x7f105ee91c50>
        param tags User assigned tags

    title = None
        param title Page title

    url = None
        param url Page url

class readme.models.ItemTag(*args, **kwargs)
    ItemTag(id, name, slug)

class readme.models.TaggedItem(*args, **kwargs)
    TaggedItem(id, tag_id, content_object_id)

class readme.models.UserProfile(*args, **kwargs)
    UserProfile(user_id, theme, can_invite, new_window, items_per_page, show_excluded)

    id
        Just to tell the TaggableManager what the primary key of this model is
```

10.2 Scrapers

exception `readme.scrapers.ParserException`

Generic exception for parsers/scrapers. Indicates that a scraper cannot succeed.

`readme.scrapers.domain_parser` (*domain*)

Decorator to register a domain specific parser

Parameters `domain` – String

Returns function

`readme.scrapers.parse` (*item*, *content_type*, *text=None*, *content=None*)

Scrape info from an item

Parameters

- **content_type** – mime type
- **text** – unicode text
- **content** – byte string
- **item** – Item

`readme.scrapers.parse_github` (*item*, *content_type*, *text*)

Reads the readme of a repo if it can find one.

Parameters

- **item** – ignored
- **content_type** – ignored
- **text** – unicode text

Returns title, article

Raises `ParserException` raised if no readme is found

`readme.scrapers.parse_web_page` (*text*)

Generic web page parser with readability. Used as a fallback.

Parameters `text` – unicode text

Returns title, article

Raises `ParserException`

About Pypo

Pypo is a self hosted bookmarking service like [Pocket](#). There also is a rudimentary android application and firefox extension to add and view the bookmarks.

It's main components are built with:

- Python 3
- Postgresql
- Django
- readability-lxml
- Whoosh
- django-haystack
- django-taggit
- tld
- South
- requests
- django-rest-framework
- py.test
- bleach

11.1 Documentation

Full documentation can be found at [readthedocs](#)

11.2 Features

- Adding links and fetch their summaries and titles
- Links can have multiple tags
- Search by title, url and tags
- Filter by tags

11.3 Installation

1. Create a virtualenv and

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

2. Setup a postgresql db

3. You can overwrite the default settings by creating a settings_local.py next to pypo/settings.py . Do not directly edit the settings.py.

4. Install js modules with bower

```
$ npm install -g bower
$ bower install
```

5. Install uglify for js and css minify

```
$ npm install -g uglify
```

6. Setup the database

```
$ ./manage.py syncdb
$ ./manage.py migrate
```

7. Add a superuser

```
$ ./manage.py createsuperuser
```

8. Host the application, see [Deploying Django with WSGI](#)

9. Create normal users with the admin interface /admin

11.4 Deploying

There is a fab file you can customize to your liking. It creates a virtualenv, sets up the directory structure and checks your current local commit out on the target machine.

11.5 Requirements status

11.6 License

This project is licensed under the terms of the Apache License version 2. See COPYING.txt for details.

Indices and tables

- *genindex*
- *search*

r

`readme.models`, [19](#)
`readme.scrapers`, [20](#)

C

`created` (`readme.models.Item` attribute), [19](#)

D

`domain_parser()` (in module `readme.scrapers`), [20](#)

F

`fetch_article()` (`readme.models.Item` method), [19](#)

I

`id` (`readme.models.UserProfile` attribute), [19](#)

`Item` (class in `readme.models`), [19](#)

`ItemTag` (class in `readme.models`), [19](#)

O

`owner` (`readme.models.Item` attribute), [19](#)

P

`parse()` (in module `readme.scrapers`), [20](#)

`parse_github()` (in module `readme.scrapers`), [20](#)

`parse_web_page()` (in module `readme.scrapers`), [20](#)

`ParserException`, [20](#)

R

`readable_article` (`readme.models.Item` attribute), [19](#)

`readme.models` (module), [19](#)

`readme.scrapers` (module), [20](#)

S

`safe_article` (`readme.models.Item` attribute), [19](#)

T

`TaggedItem` (class in `readme.models`), [19](#)

`tags` (`readme.models.Item` attribute), [19](#)

`title` (`readme.models.Item` attribute), [19](#)

U

`url` (`readme.models.Item` attribute), [19](#)

`UserProfile` (class in `readme.models`), [19](#)