django-webdav-storage

Release 1.0

Feb 18, 2020

Contents

https://github.com/marazmiki/django-webdav-storage

CHAPTER 1

Description

django-webdav-storage is pluggable Django application allows you easily save media and static files into your own WebDAV-storage.

CHAPTER 2

Table of contents

2.1 Requirements

- The package works both on Python 2.7 and 3.5+.
- Required Django version is `1.11+`. The `2.x` and `3.x` branches are also supported.
- The requests library.

Attention: In python 3.x, `ContentFile` with text mode content (not binary one) will causes `TypeError` due requests restrictions.

2.2 Installation

You can install the latest stable version of `django-webdav-storage` from PyPI within `pip` command:

```
$ pip install django-webdav-storage
```

Of course, you also can use a development version:

```
$ pip install -e git+github.com/marazmiki/django-webdav-storage.git#egg=django-webdav-
↪storage
```

All dependencies will be installed automatically

2.3 Configuration

2.3.1 Base settings

First of all, you need set the your WebDAV server url (`WEBDAV_URL` setting) and public read-only public url of this if there (`WEBDAV_PUBLIC_URL` setting)

```
# settings.py

WEBDAV_URL = "https://my-internal-webdav-server.example.com"
WEBDAV_PUBLIC_URL = "http://my-awesome-public-webdav-server.example.com"
```

If you want use HTTP Basic authorization to WebDAV access, you can specify your credentials like that:

```
WEBDAV_URL = 'http://johndoe:secret@my-internal-webdav-server.example.com'
```

Second, set the `django_webdav_storage.storage.WebDavStorage` storage class as default storage class:

```
DEFAULT_FILE_STORAGE = 'django_webdav_storage.storage.WebDavStorage'
```

If your webdav backend can't recursively create path, set the `WEBDAV_RECURSIVE_MKCOL` variable to `True` (please pay attention: nginx can do this):

```
WEBDAV_RECURSIVE_MKCOL = True     # *NOT* required for nginx!
```

Attention: The `WEBDAV_RECURSIVE_MKCOL` setting must be `False` if you use nginx as WebDAV service since `nginx` allows you automatically create full path to uloaded file when `create_full_put_path` is `on`.

2.3.2 List support in nginx

If you use **nginx** as WebDAV server and want to enable storage directory listing, set the `WEBDAV_LISTING_BACKEND` option to:

```
WEBDAV_LISTING_BACKEND = 'django_webdav_storage.listing.nginx_autoindex'
```

Autoindex feature must be enabled in your nginx configuration for application servers (see example below). Be careful! Allowing autoindex for any user may lead to security and performance issues.

Also, you may specify path to other function with the following interface:

```
def listdir(storage_object, path_string):
    return dirs_list, files_list
```

2.4 Examples

2.4.1 Add WebDAV support to nginx

```
# Public readonly media server.
server {
    listen 80;
    charset         utf-8;
    server_tokens   off;
    server_name     media.example.com;

    access_log      /var/log/nginx/media_access.log;
    error_log       /var/log/nginx/media_error.log;

    root            /usr/share/nginx/webdav;

}

# WebDAV server
server {
    listen 80;
    charset         utf-8;
    server_tokens   off;
    server_name     webdav.example.com;

    access_log      /var/log/nginx/webdav_access.log;
    error_log       /var/log/nginx/webdav_error.log;

    root            /usr/share/nginx/webdav;

    client_max_body_size    10m;
    client_body_temp_path   /tmp;
    create_full_put_path    on;
    autoindex               on;

    dav_methods             PUT DELETE MKCOL COPY MOVE;
    dav_access              user:rw   group:r   all:r;

    satisfy                 any;

    allow                   127.0.0.1/32;
    deny                    all;

    auth_basic              'My WebDAV area';
    auth_basic_user_file    /usr/share/nginx/.htpasswd;
}
```

2.5 Change log

- Using `poetry` in the development;
- Using `py.test` instead of unittest-based Django test framework;
- Testing locally in all the supported environments with tox
- Added support for Python `3.6`, `3.7` and `3.8`;
- Dropped support for Python `3.2`, `3.3` and `3.4`
- Added support for Django `1.11`, `2.x` and `3.x`;
- Dropped support of all the older Django versions;

- Added the `run_wsgidav` command that creates a developer WebDAV server;
- Added an example project;
- Improved documentation;

2.5.1 0.6x

0.6.1

- Cleanup code, remove unused code — thanks to Matvei Kroglov

0.6

- Add directory listing feature (if webdav server supports it) — thanks to Matvei Kroglov

2.5.2 0.5x

0.5

- Chunked upload support — thanks to Matvei Kroglov

2.5.3 0.4x

0.4.2

- Update django versions

0.4.1

- Update django versions

0.4

- Updated head django versions;
- Support of Django 1.8x;
- Explicit directory tree creation (MKCOL) now os optional — thanks to Dmitriy Narkevich
- Added missed self.webdav wrapper for MKCOL request

2.5.4 0.3x

0.3

- Updated head django versions;
- Explicit directory tree creation (MKCOL) — thanks to Richard Lander;
- Fix set_public_url and set_webdav_url method names — thanks to fuxter;

- Added AUTHORS.rst and CHANGELOG.rst files

2.5.5 0.2x

0.2

- Updated head Django versions.

2.5.6 0.1x

0.1

- Initial release.

2.6 Authors

Many thanks to all these people who make the `django-webdav-storage` better:

- Mikhail marazmiki Porokhovnichenko <marazmiki@gmail.com>
- Richard lander2k2 Lander <lander2k2@gmail.com>
- Dmitriy dimier Narkevich <github@dimier.org>
- Matvey subuk Kruglov <kubuzzzz@gmail.com>
- fuxter <fuxterz@gmail.com>
- Evgeniy lowitea <ea@lowit.ru>

CHAPTER 3

Tables and indicies

- genindex
- search