
Django FTP server Documentation

Release 0.7.0

Shinya Okano

Feb 23, 2022

CONTENTS

1	Django FTP server	3
1.1	Getting Started	3
1.2	Requirements	4
1.3	License	4
1.4	Documentation	4
1.5	Develop	4
1.6	Author	4
2	Management commands	5
2.1	ftpsrvr	5
2.2	createftpuseraccount	5
2.3	createftpusergroup	6
3	Custom Handlers and Authorizers	7
4	Using Django storage system (Experimental)	9
5	Reference	11
5.1	django_ftpsrvr.admin	11
5.2	django_ftpsrvr.authorizers	11
5.3	django_ftpsrvr.filesystems	12
5.4	django_ftpsrvr.models	13
5.5	django_ftpsrvr.utils	14
6	Change History	15
6.1	0.8.0	15
6.2	0.7.0	15
6.3	0.6.0	15
6.4	0.5.0	16
6.5	0.4.1	16
6.6	0.4.0	16
6.7	0.3.5	16
6.8	0.3.4	16
6.9	0.3.3	16
6.10	0.3.2	17
6.11	0.3.1	17
6.12	0.3	17
6.13	0.2	17
6.14	0.1	17
7	Indices and tables	19

Python Module Index	21
Index	23

FTP server application that used user authentication of Django.

Contents:

DJANGO FTP SERVER

FTP server application that used user authentication of Django.

1.1 Getting Started

1. Install django-ftpserver by pip.

```
$ pip install django-ftpserver
```

2. Add line to settings.INSTALLED_APPS for your django project.

```
INSTALLED_APPS = (  
    # ..  
    'django_ftpserver',  
)
```

3. Migrate app.

```
$ python manage.py migrate
```

4. Create FTP user group.

```
$ python manage.py createftpgroup my-ftp-group
```

5. Create FTP user account.

```
$ python manage.py createftpuseraccount <username> my-ftp-group
```

<username> is the django authentication username.

6. Run manage.py ftpserver command.

```
$ python manage.py ftpserver 127.0.0.1:10021
```

7. Connect with your favorite FTP client.

1.2 Requirements

- Target Python version is 3.6, 3.7, 3.8, 3.9, 3.10
- Django>=2.2
- pyftplib

1.3 License

This software is licensed under the MIT License.

1.4 Documentation

The latest documentation is hosted at Read The Docs.

<https://django-ftpserver.readthedocs.org/en/latest/>

1.5 Develop

This project is hosted at Github: <https://github.com/tokibito/django-ftpserver>

1.6 Author

- Shinya Okano

MANAGEMENT COMMANDS

Django FTP server add some commands to `manage.py` commands.

2.1 ftpserver

Start FTP server.

Usage:

```
$ python manage.py ftpserver [options] [host:port]
```

[host:port] is bind address for FTP server.

For example, it binds to address of 10.0.0.1:21:

```
$ python manage.py ftpserver 10.0.0.1:21
```

Table 1: options

Option	Description
<code>--daemonize</code>	become background service.
<code>--pidfile=PIDFILE</code>	filename to write process id (PID).
<code>--timeout=TIMEOUT</code>	timeout for remote client.
<code>--passive-ports=PASSIVE-PORTS</code>	Passive ports. eg. 12345,30000-50000
<code>--masquerade-address=MASQUERADE-ADDRESS</code>	masquerade address.
<code>--file-access-user=FILE-ACCESS-USER</code>	user for access to file.
<code>--certfile=CERTFILE</code>	TLS certificate file.
<code>--keyfile=KEYFILE</code>	TLS private key file.
<code>--sendfile</code>	Use sendfile.

2.2 createftputeraccount

Create a FTP user account (FTPUserAccount record).

Usage:

```
$ python manage.py createftputeraccount [options] <username> <group> [home_dir]
```

2.3 createftpgroup

Create a FTP user group (FTPUserGroup record).

Usage:

```
$ python manage.py createftpgroup [options] <name> [home_dir]
```

CUSTOM HANDLERS AND AUTHORIZORS

Django FTP Server lets you specify custom handlers and authorizers via Django settings.

Setting Options:

```
FTPSERVER_AUTHORIZER = 'django_ftpserver.authorizers.FTPAccountAuthorizer'  
FTPSERVER_HANDLER = 'pyftplib.handlers.FTPHandler'  
FTPSERVER_TLSHANDLER = 'pyftplib.handlers.TLS_FTPHandler'
```

The class definitions and methods can be found at the [pyftplib's documentation](#).

USING DJANGO STORAGE SYSTEM (EXPERIMENTAL)

Note: This function is experimental. The API may change frequently.

Django FTP Server supports the Django storage API.

With StorageFS, you can change the storage of the FTP server to various things.

Settings:

```
# Change FTP server filesystem
FTP_SERVER_FILESYSTEM = 'django_ftpserver.filesystems.StorageFS'
# Using Amazon S3 storage (django-storages)
DEFAULT_FILE_STORAGE = 'storages.backends.s3boto3.S3Boto3Storage'
# Config for S3Boto3Storage
AWS_ACCESS_KEY_ID = '(your access key id)'
AWS_SECRET_ACCESS_KEY = 'your secret access key'
AWS_STORAGE_BUCKET_NAME = 'your.storage.bucket'
```


REFERENCE

5.1 django_ftpserver.admin

class django_ftpserver.admin.FTPUserAccountAdmin(*model*, *admin_site*)
Admin class for FTPUserAccountAdmin

class django_ftpserver.admin.FTPUserGroupAdmin(*model*, *admin_site*)
Admin class for FTPUserGroup

5.2 django_ftpserver.authorizers

class django_ftpserver.authorizers.FTPAccountAuthorizer(*file_access_user=None*)
Authorizer class by django authentication.

get_account(*username*)
return user by username.

get_msg_login(*username*)
message for welcome.

get_perms(*username*)
return user permissions

has_perm(*username*, *perm*, *path=None*)
check user permission

has_user(*username*)
return True if exists user.

impersonate_user(*username*, *password*)
delegate to personate_user method

model
alias of *django_ftpserver.models.FTPUserAccount*

terminate_impersonation(*username*)
delegate to terminate_impersonation method

validate_authentication(*username*, *password*, *handler*)
authenticate user with password

5.3 django_ftpserver.filesystems

class django_ftpserver.filesystems.DjangoGCloudStoragePatch

StoragePatch for DjangoGCloudStorage(provided by django-gcloud-storage).

class django_ftpserver.filesystems.FileSystemStoragePatch

StoragePatch for Django's FileSystemStorage.

class django_ftpserver.filesystems.PseudoStat(*st_size, st_mtime, st_nlink, st_mode, st_uid, st_gid, st_dev, st_ino*)

st_dev

Alias for field number 6

st_gid

Alias for field number 5

st_ino

Alias for field number 7

st_mode

Alias for field number 3

st_mtime

Alias for field number 1

st_nlink

Alias for field number 2

st_size

Alias for field number 0

st_uid

Alias for field number 4

class django_ftpserver.filesystems.S3Boto3StoragePatch

StoragePatch for S3Boto3Storage(provided by django-storages).

class django_ftpserver.filesystems.StorageFS(*root, cmd_channel*)

FileSystem for bridge to Django storage.

apply_patch()

apply adjustment patch for storage

chdir(*path*)

Change the current directory. If this method is overridden it is vital that *cwd* attribute gets set.

chmod(*path, mode*)

Change file/directory mode.

get_group_by_gid(*gid*)

Return the groupname associated with group id. If this can't be determined return raw gid instead. On Windows just return "group".

get_user_by_uid(*uid*)

Return the username associated with user id. If this can't be determined return raw uid instead. On Windows just return "owner".

getmtime(*path*)

Return the last modified time as a number of seconds since the epoch.

getsize(*path*)

Return the size of the specified file in bytes.

isdir(*path*)

Return True if path is a directory.

isfile(*path*)

Return True if path is a file.

islink(*path*)

Return True if path is a symbolic link.

lexists(*path*)

Return True if path refers to an existing path, including a broken or circular symbolic link.

listdir(*path*)

List the content of a directory.

lstat(*path*)

Like stat but does not follow symbolic links.

mkdir(*path*)

Create the specified directory.

mkstemp(*suffix=""*, *prefix=""*, *dir=None*, *mode='wb'*)

A wrap around tempfile.mkstemp creating a file with a unique name. Unlike mkstemp it returns an object with a file-like interface.

open(*filename*, *mode*)

Open a file returning its handler.

realpath(*path*)

Return the canonical version of path eliminating any symbolic links encountered in the path (if they are supported by the operating system).

remove(*path*)

Remove the specified file.

rmdir(*path*)

Remove the specified directory.

stat(*path*)

Perform a stat() system call on the given path.

class django_ftpserver.filesystems.StoragePatch

Base class for patches to StorageFS.

classmethod apply(*fs*)

replace bound methods of fs.

5.4 django_ftpserver.models

class django_ftpserver.models.FTPUserAccount(*id*, *user*, *group*, *last_login*, *home_dir*)

exception DoesNotExist

exception MultipleObjectsReturned

class django_ftpserver.models.FTPUserGroup(*id*, *name*, *permission*, *home_dir*)

exception DoesNotExist

exception MultipleObjectsReturned

5.5 django_ftpserver.utils

`django_ftpserver.utils.get_settings_value(name)`

Return the django settings value for name attribute

`django_ftpserver.utils.make_server(server_class, handler_class, authorizer_class, filesystem_class,
host_port, file_access_user=None, **handler_options)`

make server instance

Host_port (host, port)

File_access_user 'spam'

handler_options:

- timeout
- passive_ports
- masquerade_address
- certfile
- keyfile

`django_ftpserver.utils.parse_ports(ports_text)`

Parse ports text

e.g. ports_text = "12345,13000-15000,20000-30000"

CHANGE HISTORY

6.1 0.8.0

release-date 2022-02-24

- Added Django 3.1, 3.2, 4.0 support
- Added Python 3.9, 3.10 support
- Removed Django <2.2 support
- Removed Python 3.4, 3.5 support
- Enable encryption before authentication for FTP+TLS [#21](#)

6.2 0.7.0

release-date 2020-02-20

- Added Django 3.0 support
- Added Python 3.8 support
- Removed Django <2.0 support
- Removed Python 2.7 support
- Removed six dependency

6.3 0.6.0

release-date 2018-11-26

- Added Django 2.0, 2.1 support
- Removed Django <1.11 support
- Change the test runner from nose to pytest

6.4 0.5.0

release-date 2017-05-24

- Added storage system support [#14](#)
- Added Django 1.11 support

6.5 0.4.1

release-date 2017-03-27

- Update daemonize.py [#13](#)
- Add six module to install_require
- Added Python 3.6 support

6.6 0.4.0

release-date 2016-09-16

- WIP: Added django 1.10 support [#12](#)
- Removed older Python(2.6, 3.3) and Django(<1.8) support

6.7 0.3.5

release-date 2016-01-26

- Fix daemonize problem in Django 1.9 [#10](#)

6.8 0.3.4

release-date 2015-12-15

- add tox env for Django 1.9 and Python 3.5

6.9 0.3.3

release-date 2015-10-14

- [#9](#) Fix for python3 in utils

6.10 0.3.2

release-date 2015-10-02

- #7 support Custom Authorizer and Handler classes via settings

6.11 0.3.1

release-date 2015-03-29

- small refactoring

6.12 0.3

release-date 2015-2-12

- support sendfile (`--sendfile` option)
- fixes #5 support custom User username field
- fixes #4 support Windows platform
- fixes #1 model string format

6.13 0.2

release-date 2014-03-26

- support TLS (`--certfile` option)
- testing on Python 3.4

6.14 0.1

release-date 2014-03-09

first release.

INDICES AND TABLES

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

PYTHON MODULE INDEX

d

- `django_ftpserver.admin`, 11
- `django_ftpserver.authorizers`, 11
- `django_ftpserver.filesystems`, 12
- `django_ftpserver.models`, 13
- `django_ftpserver.utils`, 14

INDEX

A

`apply()` (*django_ftpserver.filesystems.StoragePatch*
class method), 13

`apply_patch()` (*django_ftpserver.filesystems.StorageFS*
method), 12

C

`chdir()` (*django_ftpserver.filesystems.StorageFS*
method), 12

`chmod()` (*django_ftpserver.filesystems.StorageFS*
method), 12

D

`django_ftpserver.admin`
module, 11

`django_ftpserver.authorizers`
module, 11

`django_ftpserver.filesystems`
module, 12

`django_ftpserver.models`
module, 13

`django_ftpserver.utils`
module, 14

`DjangoGCloudStoragePatch` (class in
django_ftpserver.filesystems), 12

F

`FileSystemStoragePatch` (class in
django_ftpserver.filesystems), 12

`FTPAccountAuthorizer` (class in
django_ftpserver.authorizers), 11

`FTPUserAccount` (class in *django_ftpserver.models*), 13

`FTPUserAccount.DoesNotExist`, 13

`FTPUserAccount.MultipleObjectsReturned`, 13

`FTPUserAccountAdmin` (class in
django_ftpserver.admin), 11

`FTPUserGroup` (class in *django_ftpserver.models*), 13

`FTPUserGroup.DoesNotExist`, 13

`FTPUserGroup.MultipleObjectsReturned`, 13

`FTPUserGroupAdmin` (class in *django_ftpserver.admin*),
11

G

`get_account()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

`get_group_by_gid()` (*django_ftpserver.filesystems.StorageFS*
method), 12

`get_msg_login()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

`get_perms()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

`get_settings_value()` (in module
django_ftpserver.utils), 14

`get_user_by_uid()` (*django_ftpserver.filesystems.StorageFS*
method), 12

`getmtime()` (*django_ftpserver.filesystems.StorageFS*
method), 12

`getsize()` (*django_ftpserver.filesystems.StorageFS*
method), 12

H

`has_perm()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

`has_user()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

I

`impersonate_user()` (*django_ftpserver.authorizers.FTPAccountAuthorizer*
method), 11

`isdir()` (*django_ftpserver.filesystems.StorageFS*
method), 12

`isfile()` (*django_ftpserver.filesystems.StorageFS*
method), 13

`islink()` (*django_ftpserver.filesystems.StorageFS*
method), 13

L

`lexists()` (*django_ftpserver.filesystems.StorageFS*
method), 13

`listdir()` (*django_ftpserver.filesystems.StorageFS*
method), 13

`lstat()` (*django_ftpserver.filesystems.StorageFS*
method), 13

M

`make_server()` (in module `django_ftpserver.utils`), 14
`mkdir()` (`django_ftpserver.filesystems.StorageFS` method), 13
`mkstemp()` (`django_ftpserver.filesystems.StorageFS` method), 13
`model` (`django_ftpserver.authorizers.FTPAccountAuthorizer` attribute), 11
`module`
 `django_ftpserver.admin`, 11
 `django_ftpserver.authorizers`, 11
 `django_ftpserver.filesystems`, 12
 `django_ftpserver.models`, 13
 `django_ftpserver.utils`, 14

O

`open()` (`django_ftpserver.filesystems.StorageFS` method), 13

P

`parse_ports()` (in module `django_ftpserver.utils`), 14
`PseudoStat` (class in `django_ftpserver.filesystems`), 12

R

`realpath()` (`django_ftpserver.filesystems.StorageFS` method), 13
`remove()` (`django_ftpserver.filesystems.StorageFS` method), 13
`rmdir()` (`django_ftpserver.filesystems.StorageFS` method), 13

S

`S3Boto3StoragePatch` (class in `django_ftpserver.filesystems`), 12
`st_dev` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_gid` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_ino` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_mode` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_mtime` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_nlink` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_size` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`st_uid` (`django_ftpserver.filesystems.PseudoStat` attribute), 12
`stat()` (`django_ftpserver.filesystems.StorageFS` method), 13
`StorageFS` (class in `django_ftpserver.filesystems`), 12

`StoragePatch` (class in `django_ftpserver.filesystems`), 13

T

`terminate_impersonation()`
 (`django_ftpserver.authorizers.FTPAccountAuthorizer` method), 11

V

`validate_authentication()`
 (`django_ftpserver.authorizers.FTPAccountAuthorizer` method), 11