Index

msaFilesystem - Agnostic Abstract Filesystem API which allows to use S3, GCS, Azure Datalake, your local FS, Youtube etc

pypi package package or version not found python package or version not found

Features

- **based on**: PyFilesystem2
- **App Filesystems**: Manage filesystems in platform-specific application directories. These classes abstract away the different requirements for user data across platforms.
- **FTP Filesystem**: A FTP (File Transport Protocol) Filesystem. Optionally, the connection can be made securely via TLS. This is known as FTPS, or FTP Secure.
- **Memory Filesystem**: A filesystem that stored in memory. Memory filesystems are useful for caches, temporary data stores, unit testing, etc.
- **Mount Filesystem**: A Mount FS is a virtual filesystem which can seamlessly map sub-directories on to other filesystems.
- **Multi Filesystem**: A MultiFS is a filesystem composed of a sequence of other filesystems, where the directory structure of each overlays the previous filesystem in the sequence.
- **OS Filesystem**: Manage the filesystem provided by your OS. In essence, an OSFS is a thin layer over the io and os modules of the Python standard library.

- **Sub Filesystem**: Manage a directory in a parent filesystem. A SubFS is a filesystem object that maps to a sub-directory of another filesystem.
- **Tar Filesystem**: Read and write tar files.
- **Temporary Filesystem**: Manage filesystems in temporary locations. A temporary filesytem is stored in a location defined by your OS (/tmp on linux). The contents are deleted when the filesystem is closed.
- **Zip Filesystem**: Read and write zip files.
- **SMB Filesystem**: A filesystem over SMB.
- **WebDAV Filesystem**: A filesystem for WebDAV.
- **Azure Datalake & S3FS Filesystem**: A filesystem for Azure Datalake storage & S3FS.
- **Google Cloud Storage (GCS) Filesystem**: A filesystem for Google Cloud Storage (GCS). With GCSFS, you can interact with Google Cloud Storage as if it was a regular filesystem.
- **Google Drive Filesystem**: A filesystem for Google Drive. Interact with Google Drive as if it was a regular filesystem.
- **Dropbox Filesystem**: A filesystem for Dropbox.
- **OneDrive Filesystem**: A filesystem for Dropbox.
- **YouTube Videos and Playlists Filesystem**: A filesystem for YouTube Videos and Playlists.
- **External Filesystems**: See the following wiki page for a list of filesystems not in the core library, and community contributed filesystems.

Main Dependencies

- **fs~=2.4.16**: Module that provides a common interface to any filesystem
- **six~=1.16.0**: Python 2 and 3 compatibility utilities
- **pysmb~=1.2.8**: SMB/CIFS library
- **fs.webdavfs~=0.4.2**: WebDAV support
- **fs-dlk~=0.1.3**: Azure Datalake support
- **fs-s3fs~=1.1.1**: Amazon S3 filesystem support
- **fs-gcsfs~=1.5.1**: Google Cloud Storage (GCS) support
- **fs.googledrivefs~=2.3.0**: Google Drive support
- **fs.dropboxfs~=0.2.2.post2**: Dropbox support
- **fs.onedrivefs~=1.1.1**: OneDrive support
- **fs.youtube~=0.3.1**: Youtube support
- **fs.smbfs~=1.0.5**: SMB support

Usage example

```python
from msaFilesystem.msafs import MSAFilesystem
from fs.walk import Walker

# FS URLs are formatted in the following way:
# <protocol>://<username>:<password>@<resource>

myFS = MSAFilesystem(fs_url='osfs://~/projects')

walker = Walker(filter=['*.py'])
for path in walker.files(myFS.fs):
    print(path)
```

License Agreement

- `msaFilesystem` is based on `MIT` open source and free to use, it is free for commercial use, but please show/list the copyright information about msaFilesystem somewhere.

How to create the documentation

We use mkdocs and mkdocsstring. The code reference and nav entry get's created virtually by the triggered python script /docs/gen_ref_pages.py while `mkdocs` `serve` or `build` is executed.

Requirements Install for the PDF creation option:

PDF Export is using mainly weasyprint, if you get some errors here pls. check there documentation. Installation is part of the msaFilesystem, so this should be fine.

We can now test and view our documentation using:

```
mkdocs serve
```

Build static Site:

```
mkdocs build
```

Build and Publish

Build:

```
python setup.py sdist
```

Publish to pypi:

```
twine upload dist/*
```