django-selectel-storage

Release 1.0.2

May 05, 2020

Contents:

Introduction

1.1 What is "django-selectel-storage"?

It's an implementation of Django Storage API to seamless store all the data inside the Selectel Cloud Storage.

1.2 What is "Django"?

It's funny. Of course, you know Django. And know about storage API in Django. Nothing to say here.

1.3 What is "Selectel"?

Selectel is one of the leading Russian internet hosting operators. Besides a huge number of useful services it provided, would like to say a couple words about Cloud Storage. It does the same as Amazon's S3, but simpler and cheaper. . . a bit.

1.4 Why "Selectel" cloud?

Well, clouds are so cute. Always use it. Also, if your business is located in Russia, you should store sensitive privacy data in Russia, according to some stupid legal acts. Probably, Selectel Cloud Storage is the best choice, because it's relatively cheap and reliable.

Installation

2.1 Get a stable version

with `pip`:

```
$ pip install django-selectel-storage
```

or via pipenv:

```
$ pipenv install django-selectel-storage
```

or even via poetry:

```
$ poetry add django-selectel-storage
```

2.2 Get a developer version

Just install it from github also via `pip`

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

You even get a code of mandatory git revision:

```
$ pip install -e git+https://github.com/marazmiki/django-selectel-storage@{REV_HASH}
↪#egg=django-selectel-storage
```

Configuration

Here, we assume you already:

- Have registered an account on Selectel,
- Spent some money there,
- Created a couple of containers and got credentials for it.

if you haven't yet, consider doing that right now.

3.1 A minimal installation

To begin using Selectel storage as your storage, you should configure your containers credentials:

```
SELECTEL_STORAGES = {
    'default': {
        'USERNAME': 'xxxx_user',
        'PASSWORD': 'p455w0rd',
        'CONTAINER': 'my-data',
    },
    'yet_another_storage': 'selectel://xxxx_user:p455w0rd2@another-container',
}
```

As you already guessed, the dictionary means you can create as much configurations as you want. Here, we have two schemas: `default` and `yet_another_storage`.

You're free to use any names for schema but strongly recommended to declare a schema with name ''default''.

Also, as you already guessed again, you can specify credentials either:

- As a dictionary with `USERNAME`, `PASSWORD` or `CONTAINER` keys, or
- a URL-like string with schema `selectel://`

Please see all available configuration options below.

> **Attention:** Credentials in examples above are hardcoded. It really sucks, don't do that in real life. Instead, you can store these variables in the environment and extract it out of there, for example, via awesome python-decouple package.

3.2 Using django-selectel-storage as a default backend

If you want to use the storage by default, consider adding to your `settings.py`:

```
DEFAULT_FILE_STORAGE = 'django_selectel_storage.storage.SelectelStorage'
```

in this case, the `default` schema will be implicitly used. Please make sure you defined it before.

3.3 Storage initialization

If you want to explicitly instantiate a storage, you should write something like that:

```python
# models.py
from django.db import models


class Post(models.Model):
    photo = models.ImageField(storage=storage_instance)
```

Wondering, what `storage_instance` is? Well, it would be one of:

```python
from django_selectel_storage.storage import SelectelStorage

storage_instance = SelectelStorage()
storage_instance = SelectelStorage('default')
storage_instance = SelectelStorage(storage='default')
```

Please pay attention: the last three lines of code do the same thing: creates a `SelectelStorage` instance with using `default` schema. Of course, you can choose any different schema (if you've defined it of course):

```python
storage_instance = SelectelStorage('yet_another_storage')
storage_instance = SelectelStorage(storage='yet_another_storage')
```

We even create an instance via DSN (it's useful somewhere in `./manage.py shell`, please don't do that in production):

```python
storage_instance = SelectelStorage('selectel://user:password@container_name/')
storage_instance = SelectelStorage(storage='selectel://user:password@container_name/')
```

An interesting gotcha here: if you specify a DSN (a string begins with `selectel://`) as a first positional argument, it would act like a dsn rather as `storage=`

Changelog

4.1 1.x

4.2 1.0.2

- A new feature `sendmefile` implementation to make direct upload to a container.
- Fixed an issue when authentication gets disappeared in some cases (#10, thanks to @idealatom for PR);
- Fixed some misspells in docs;

4.3 1.0.1

- Fixed an issue when accessing a container without explicit authentication (thanks to Alexey Kotenko for reporting)
- Added `requests` as a required dependency.

4.3.1 1.0

- Added support for Python `3.6`, `3.7` and `3.8`; in the opposite, dropped support for old ones (`3.4` and lower)
- Dropped support for older `Django` versions (the oldest one we supporting is `1.10`)
- A new configuration format allowing create a number of different schemas;
- Get rid off the 3rd party dependency: `selectel-api`;
- Using `tox` and `pytest` utilities when developing and testing;
- Using `poetry` as package management and deploying tool, so `setup.py` is no longer needed;

- All the development utils configs (such as `.rccoverage`, `tox.ini` and so on) also moved in the only `setup.cfg` file

- License is MIT

4.4 0.3x

4.4.1 0.3.1

Released at 2016-03-13

- Fix static storage settings

4.4.2 0.3.0

Released at 2016-02-07

- Drop older python verions support (vv 3.2, 3.3)

- Drop older Django version support (1.6)

- Use tox for testing

4.5 0.2x

4.5.1 0.2.2

2015-05-23

- Update django head versions

4.5.2 0.2.1

2015-05-02

- Update head django version

4.5.3 0.2.0

2015-04-04

- 5ac502a 2014-10-03 | init? [Mikhail Porokhovnichenko]

Indices and tables

- genindex
- modindex
- search