
django-gaekit Documentation

Release 1.0.2

George Whewell

April 27, 2015

| | | |
|----------|------------------------------|-----------|
| 1 | django-gaekit | 3 |
| 1.1 | Documentation | 3 |
| 1.2 | Features | 3 |
| 2 | Installation | 5 |
| 3 | Usage | 7 |
| 3.1 | Storage Backend | 7 |
| 3.2 | Cache | 7 |
| 3.3 | Blacklist | 7 |
| 4 | Configuration | 9 |
| 5 | Credits | 11 |
| 5.1 | Development Lead | 11 |
| 5.2 | Contributors | 11 |
| 6 | History | 13 |
| 6.1 | 1.0.2 (2015-04-27) | 13 |
| 6.2 | 1.0.1 (2015-03-24) | 13 |
| 6.3 | 1.0.0 (2015-02-18) | 13 |
| 6.4 | 0.2.9 (2014-08-13) | 13 |
| 6.5 | 0.2.8 (2014-08-13) | 13 |
| 6.6 | 0.2.7 (2014-04-17) | 14 |
| 6.7 | 0.1.4 (2014-03-10) | 14 |
| 6.8 | 0.1.3 (2014-03-10) | 14 |
| 6.9 | 0.1.2 (2014-03-07) | 14 |
| 6.10 | 0.1.1 (2013-10-10) | 14 |
| 6.11 | 0.1.0 (2013-08-15) | 14 |

Contents:

django-gaekit

Collection of backends, wrappers and utilities to enquicken django development on Google App Engine

1.1 Documentation

The full documentation is at <http://django-gaekit.rfd.org>.

1.2 Features

- Django storage backend using Google Cloud Storage
- Django cache backend using Memcache
- Import blacklisted modules in the SDK (eg sqlite3)

Installation

To use with virtualenv, add the following to your requirements.txt:

```
django-gaekit==0.2.9
```

As App Engine does not support installing requirements automatically, you need to manually install dependencies to your project folder and upload them with your app. One way to do this is with pip's `-target` option:

```
$ pip install -t libs -r requirements.txt
```

You then need to add the libs folder to your `sys.path` before libraries are loaded. You can use `appengine_config.py` for this, since it's loaded on startup in the SDK and on production environment:

```
import sys
sys.path.insert(0, 'path/to/lib')
```

You will also want to import this file (or copy the above) in your `manage.py`, so libraries are loaded for running management commands:

3.1 Storage Backend

To use the storage backend, add the following to your settings module:

```
DEFAULT_FILE_STORAGE = 'gaekit.storages.CloudStorage'  
GS_BUCKET_NAME = 'your_project_name'
```

3.2 Cache

To use the cache backend, add the following to your settings module:

```
CACHES = {  
    'default': {  
        'BACKEND': 'gaekit.caches.GAEMemcachedCache',  
    }  
}
```

3.3 Blacklist

To import blacklisted modules, in your **local** settings module:

```
from gaekit.boot import break_sandbox  
break_sandbox()
```

Configuration

`IMAGESERVICE_UPLOAD_HEADERS` sets the headers that are set when the file is uploaded to cloud storage. This defaults to “`{ 'x-goog-acl': 'public-read' }`”, which will make your files world-readable.

Credits

5.1 Development Lead

- George Whewell <george@bynd.com>

5.2 Contributors

- Salvador Faria <salvador@bynd.com>
- Paul Mison <paul@bynd.com>

6.1 1.0.2 (2015-04-27)

- Adds wagtail.wagtailimages integration tests
- Storage.open now raises IOError if file doesn't exist
- Storage.size now raises OSError if file doesn't exist

6.2 1.0.1 (2015-03-24)

- Adds support for retrieving file size

6.3 1.0.0 (2015-02-18)

- Remove dependency on ImageService. URLs are now constructed from bucket name, with no RPC calls.
- Resolved inconsistencies with returning bucket name as part of path. This is now hidden.
- Deprecated ImageService-specific options. `IMAGESERVICE_DEFAULT_SIZE` and `IMAGESERVICE_SECURE_URLS`.
- Deprecated `drf_extensions` module.

6.4 0.2.9 (2014-08-13)

- Break out options `IMAGESERVICE_DEFAULT_SIZE`, `IMAGESERVICE_SECURE_URLS` and `IMAGESERVICE_UPLOAD_HEADERS`

6.5 0.2.8 (2014-08-13)

- Duff release

6.6 0.2.7 (2014-04-17)

- Add FeedListKeyConstructor for drf_extensions

6.7 0.1.4 (2014-03-10)

- Add fallback for cloud storage urls

6.8 0.1.3 (2014-03-10)

- Add exception logging if URL generation fails

6.9 0.1.2 (2014-03-07)

- Remove utils (lock, memoizer)
- Monkeypatch ReadBuffer to add open() stub (fixes issue with easy_thumbnails)

6.10 0.1.1 (2013-10-10)

- Fix for distributed lock

6.11 0.1.0 (2013-08-15)

- First release on PyPI.