
django-auth-iam Documentation

Release 0.1.2

Michael Budde

Aug 02, 2017

Contents

1	Installation	1
2	Usage	3
3	Settings	5
4	API	7
5	Indices and tables	9
	Python Module Index	11

CHAPTER 1

Installation

django-auth-iam can be installed with pip:

```
$ pip install django-auth-iam
```

Your Amazon credentials need to be specified in a configuration file that looks like this:

```
[Credentials]
aws_access_key_id = AKEIAJLXJFEXAMPLE
aws_secret_access_key = TLJASY/(ASF+fasdAJIdfWLasJfljaeisljae

[DB]
db_name = my_example_user_domain
```

Users will be stored in SimpleDB in the domain specified by db_name. This file can be saved where ever you want. To tell boto where it can find this file you need to set the BOTO_CONFIG environment variable. This can be done in your settings.py file:

```
import os
os.environ['BOTO_CONFIG'] = '/path/to/your/boto.cfg'
```

In your Django configuration you also need to set AUTHENTICATION_BACKEND:

```
AUTHENTICATION_BACKEND = (
    'django_auth_iam.backends.AmazonIAMBackend'
)
```

Requirements

django_auth_iam currently depends on the development version of boto. You can install this with the following command:

```
$ pip install -e git+git://github.com/boto/boto.git#egg=boto
```

Creating users

`django_auth_iam` is not integrated with the Django admin interface. You can instead use a python shell or script to create your users. You can start a python shell with `python manage.py shell`:

```
>>> from django_auth_iam.models import User
>>> User.create('user1', 'password')
User<...>
>>> User.create('user2', 'foo1234')
User<...>
>>> user = User.get_by_username('user1')
>>> user.delete()
```

Passwords are automatically hashed before they are stored. To change a users password you can use the method `change_password()`:

```
>>> user = User.create('testuser', 'pass')
>>> user.password == 'pass'
True
>>> user.change_password('pass', 'spam')
>>> print user.password
$2a$12$hmYnBI/VdPjxZep1lbIcLObB1N.LYYXRanL/1AMY1aJeIn30aBOjO
>>> user.password == 'spam'
True
```


`django-auth-iam` has the following settings that control how it works.

IAM_USER_CLASS

Default `'django_auth_iam.models.User'` (`django_auth_iam.models.User`)

Controls what class that is used when instantiating a IAM user object. This should either be the default value or point a subclass of the default class. You can use this to add extra attributes to your user model:

```
import django_auth_iam.models
from boto.sdb.db.property import *

class MyUser(django_auth_iam.models.User):

    favorite_color = StringProperty()
```

And in your `settings.py`:

```
IAM_USER_CLASS = 'myapp.models.MyUser'
```

See also:

The API documentation for `boto` contains the documentation of the different property types.

Models

Utilities

`django_auth_iam.utils.encrypt_key` (*plain, password*)

`django_auth_iam.utils.decrypt_key` (*cipher, password*)

CHAPTER 5

Indices and tables

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

d

`django_auth_iam.models`, [7](#)

`django_auth_iam.utils`, [7](#)

D

`decrypt_key()` (in module `django_auth_iam.utils`), [7](#)
`django_auth_iam.models` (module), [7](#)
`django_auth_iam.utils` (module), [7](#)

E

`encrypt_key()` (in module `django_auth_iam.utils`), [7](#)