
django-mongo-auth Documentation

Release 0.1.2

wlan slovenija

Sep 27, 2017

Contents

1	Contents	3
1.1	Installation	3
2	Source Code and Issue Tracker	5
3	Indices and tables	7

Django authentication based on an extensible [MongoEngine](#) user class.

It supports common authentication providers (Facebook, Google, Twitter, Foursquare, BrowserID/Persona, ...) and a traditional on-site registration workflow with e-mail address confirmation. Additionally, anonymous users are given a temporary account instance which can then be converted to an authenticated one. Each account can be linked with multiple authentication providers.

Installation

Using `pip` simply by doing:

```
pip install django-mongo-auth
```

You should then add `mongo_auth` and dependency `django_browserid` to `INSTALLED_APPS`. Add `django_browserid.context_processors.browserid_form` to `TEMPLATE_CONTEXT_PROCESSORS` to conclude `django_browserid` installation. Add `mongo_auth.middleware.LazyUserMiddleware` just after `django.contrib.auth.middleware.AuthenticationMiddleware` in `MIDDLEWARE_CLASSES`.

Optionally, to use provided templates, you can add `mongo_auth.contrib` and `sekizai` to `INSTALLED_APPS`, and `mongo_auth.contrib.context_processors.mongo_auth` and `sekizai.context_processors.sekizai` to `TEMPLATE_CONTEXT_PROCESSORS`, too.

Afterwards, you configure authentication providers you want to offer:

```
AUTHENTICATION_BACKENDS = (
    'mongo_auth.backends.MongoEngineBackend',
    'mongo_auth.backends.FacebookBackend',
    'mongo_auth.backends.TwitterBackend',
    'mongo_auth.backends.FoursquareBackend',
    'mongo_auth.backends.GoogleBackend',
    'mongo_auth.backends.BrowserIDBackend',
    'mongo_auth.backends.LazyUserBackend',
)
```

Some require API keys by providers. Available settings:

- `FACEBOOK_APP_ID`
- `FACEBOOK_APP_SECRET`
- `TWITTER_CONSUMER_KEY`

- `TWITTER_CONSUMER_SECRET`
- `FOURSQUARE_CLIENT_ID`
- `FOURSQUARE_CLIENT_SECRET`
- `GOOGLE_CLIENT_ID`
- `GOOGLE_CLIENT_SECRET`

If you want to use custom User class (like `mongo_auth.contrib.models.User`), you can set `USER_CLASS` to it. Default is `mongo_auth.models.User`.

`DEFAULT_USER_IMAGE` can be used to configure user image for users without one. By default it is `mongo_auth/images/unknown.png`.

Because `django.contrib.sites` does not work with `MongoEngine`, you can use `SITE_NAME` and `DEFAULT_REQUEST` to configure what site name is displayed and manually how full URLs are generated, respectively.

Add to project's `urls.py`:

```
url(r'^$', include('mongo_auth.contrib.urls')),
```


CHAPTER 2

Source Code and Issue Tracker

For development [GitHub](#) is used, so source code and issue tracker is found [there](#).

CHAPTER 3

Indices and tables

- `genindex`
- `search`