User Service Documentation

Release 0.0.1

SOON_

October 14, 2015

Contents

User service is Django app handles user authentication and authorization with 3rd party services i.e. Facebook, Google, Instagram and Twitter.

CHAPTER 1

Dev packages:

1.1 Userservice

User service is Django app handles user authentication and authorization with 3rd party services i.e. Facebook, Google, Instagram and Twitter.

1.1.1 userservice.app package

Subpackages

userservice.app.tests package

Submodules

userservice.app.tests.test_signup module

Module contents

Submodules

userservice.app.adapter module

userservice.app.admin module

userservice.app.sdk module

class `userservice.app.sdk.`**`EmailService`**(*obj*)
Bases: `object`

Abstract method for sending emails

`prepare_payload`()
Parse self.obj to another object which is understandable for an email service

`send`()
Send a request to an email service

class userservice.app.sdk.**VerificationEmail**(*obj*)
Bases: *userservice.app.sdk.EmailService*

set_body(*object*)

userservice.app.views module

Module contents

1.1.2 Settings

Base

This module is based and should not be used directly. It's intended to load middlewares, installed apps, configure another module like Django REST framework, etc.

Development

Used for local testing. Module shouldn't use any external services like storages or shared cached.

QA

Used for QA the module is most similar to production environment.

Test

Used for testing - no cache, simple password hashing, temporary file as media root, ...

CHAPTER 2

Api docs:

2.1 Api docs

The user service uses *django-allauth* and *django-rest-auth*. All endpoints are generated by *django-rest-auth* and some additional logic is applied.

The api uri is *api/v1/* and only one oauth privider is exposed as default *api/v1/facebook*

2.1.1 Examples

Getting user details without authorization token:

```
http http://localdocker:8000/api/v1/user/

HTTP/1.0 401 UNAUTHORIZED
Allow: GET, PUT, PATCH, HEAD, OPTIONS
Content-Type: application/json
Date: Tue, 13 Oct 2015 16:08:37 GMT
Server: WSGIServer/0.1 Python/2.7.10
Vary: Accept
WWW-Authenticate: Token
X-Frame-Options: SAMEORIGIN

{
    "detail": "Authentication credentials were not provided."
}
```

Send the same request but with valid authorization token:

```
http http://localdocker:8000/api/v1/user/  "Authorization: Token 04ad1f8c93bfbdd70ad21bff1ba8c21

HTTP/1.0 200 OK
Allow: GET, PUT, PATCH, HEAD, OPTIONS
Content-Type: application/json
Date: Tue, 13 Oct 2015 16:07:39 GMT
Server: WSGIServer/0.1 Python/2.7.10
Vary: Accept
X-Frame-Options: SAMEORIGIN

{
    "email": "user@mail.com",
    "first_name": "user",
```

```
        "last_name": "user",
        "username": "user"
    }
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

u

E

P

S

U

V