django_nopassword Documentation

Release 0.6.0

Rolf Erik Lekang

Sep 14, 2018

Contents

CHAPTER 1

Installation

Run this command to install django-nopassword:

```
pip install django-nopassword
```

Requirements: Django >= 1.11 (custom user is supported)

CHAPTER 2

Usage

Add the app to installed apps:

```
INSTALLED_APPS = (
    'nopassword',
)
```

Add the authentication backend *EmailBackend*:

```
AUTHENTICATION_BACKENDS = (
    # Needed to login by username in Django admin, regardless of `nopassword`
    'django.contrib.auth.backends.ModelBackend',

    # Send login codes via email
    'nopassword.backends.email.EmailBackend',
)
```

Add urls to your *urls.py*:

```
urlpatterns = patterns('',
    url(r'^accounts/', include('nopassword.urls')),
)
```

2.1 Backends

There are several predefined backends. Usage of those backends are listed below.

class `nopassword.backends.email.`**`EmailBackend`**

Delivers the code by email. It uses the django send email functionality to send the emails.

Override the following templates to customize emails:

- `registration/login_email.txt` - Plain text message

- `registration/login_email.html` - HTML message (note that no default html message is attached)
- `registration/login_subject.txt` - Subject

class nopassword.backends.sms.**TwilioBackend**

Delivers the code by sms sent through the twilio service.

Override the following template to customize messages:

- `registration/login_sms.txt` - SMS message

2.1.1 Custom backends

In backends.py there is a *NoPasswordBackend*, from which it is possible to build custom backends. The *EmailBackend* described above inherits from this backend. Creating your own backend can be done by creating a subclass of *NoPasswordBackend* and implementing *send_login_code*.:

```
class CustomBackend(NoPasswordBackend):

    def send_login_code(self, code, context, **kwargs):
        """
        Use code.user to get contact information
        Use context to render a custom template
        Use kwargs in case you have a custom view that provides additional configuration
        """
```

CHAPTER 3

REST API

To use the REST API, *djangorestframework* must be installed:

```
pip install djangorestframework
```

Add rest framework to installed apps:

```
INSTALLED_APPS = (
    ...
    'rest_framework',
    'rest_framework.authtoken',
    'nopassword',
    ...
)
```

Add *TokenAuthentication* to default authentication classes:

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework.authentication.TokenAuthentication',
    )
}
```

Add urls to your *urls.py*:

```
urlpatterns = patterns('',
    ...
    url(r'^api/accounts/', include('nopassword.rest.urls')),
    ...
)
```

You will have the following endpoints available:

- ***/api/accounts/login/*** **(POST)**
  - username

  - next (optional, will be returned in `/api/accounts/login/code/` to be handled by the frontend)
  - Sends a login code to the user
- ***/api/accounts/login/code/*** **(POST)**
  - code
  - Returns `key` (authentication token) and `next` (provided by `/api/accounts/login/`)
- ***/api/accounts/logout/*** **(POST)**
  - Performs logout

CHAPTER 4

Settings

4.1 django-nopassword settings

django.conf.settings.**NOPASSWORD_LOGIN_CODE_TIMEOUT**
Default: `900`

Defines how long a login code is valid in seconds.

django.conf.settings.**NOPASSWORD_HASH_ALGORITHM**
Default: `'sha256'`

Set the algorithm for used in logincode generation. Possible values are those who are supported in hashlib. The value should be set as the name of the attribute in hashlib. Example *hashlib.sha256()* would be ‘NOPASSWORD_HASH_ALGORITHM = ‘sha256’.

django.conf.settings.**NOPASSWORD_LOGIN_ON_GET**
Default: `False`

By default, the login code url requires a POST request to authenticate the user. A GET request renders a form that must be submitted by the user to perform authentication. To authenticate directly inside the initial GET request instead, set this to `True`.

django.conf.settings.**NOPASSWORD_CODE_LENGTH**
Default: `20`

The length of the code used to log people in.

django.conf.settings.**NOPASSWORD_TWILIO_SID**
Account ID for Twilio.

django.conf.settings.**NOPASSWORD_TWILIO_AUTH_TOKEN**
Account secret for Twilio

django.conf.settings.**NOPASSWORD_NUMERIC_CODES**
Default: `False`

A boolean flag if set to True, codes will contain numeric characters only (0-9).

4.2 Django settings used in django-nopassword

`django.conf.settings.`**`DEFAULT_FROM_EMAIL`**

Default: `'root@example.com'`

CHAPTER 5

Changelog

5.1 4.0.0

Added:

- Added `LoginCodeAdmin`
- Added rest support

Breaking changes:

- Remove support for Django < 1.11
- Add support for Django 2
- `NoPasswordBackend.authenticate` doesn't have side effects anymore, it only checks if a login code is valid.
- `NoPasswordBackend` now uses the default django method `user_can_authenticate` instead of `verify_user`.
- Changed signature of `NoPasswordBackend.send_login_code` to `send_login_code(code, context, **kwargs)`, to support custom template context.
- `EmailBackend` doesn't attach a html message to the email by default. You can provide a template `registration/login_email.html` to do so.
- Removed setting `NOPASSWORD_LOGIN_EMAIL_SUBJECT` in favor of template `registration/login_subject.txt`
- Renamed form `AuthenticationForm` to `LoginForm`
- `LoginForm` (previously `AuthenticationForm`) doesn't have side effects anymore while cleaning.
- `LoginForm` (previously `AuthenticationForm`) doesn't check for cookie support anymore.
- Removed methods `get_user` and `get_user_id` from `LoginForm` (previously `AuthenticationForm`).

- Removed method login_url and send_login_code from LoginCode (previously AuthenticationForm).
- Renamed template registration/login.html to registration/login_form.html.
- Changed content of default templates.
- Removed views login_with_code_and_username.
- Refactored views to be class based views and to use forms instead of url parameters.
- Changed url paths
- Removed setting NOPASSWORD_POST_REDIRECT, use NOPASSWORD_LOGIN_ON_GET instead.
- Removed setting NOPASSWORD_NAMESPACE.
- Removed setting NOPASSWORD_HIDE_USERNAME.
- Removed setting NOPASSWORD_LOGIN_EMAIL_SUBJECT.

Index