django-howl

unknown

Oct 15, 2020

CONTENTS:

CHAPTER

ONE

WHAT IS DJANGO-HOWL

`django-howl` is a Django app where you can add custom observers and use them to check almost everything you want and pushes alerts to signals. You can connect to the signals and handle it in your way like sending notifications over different APIs.

CHAPTER TWO

REQUIREMENTS

django-howl supports Python 3 only and requires at least Django 1.11. If you need support for Django 1.8.x or 1.9.x have a look at django-howl < 1.0.0

2.1 Installation

- Install `django-howl` (or download from PyPI):

```
pip install django-howl
```

- After Installation add it to `INSTALLED_APPS` in `settings.py`:

```
INSTALLED_APPS = (
    # other apps
    'howl',
)
```

2.2 Usage

2.2.1 Operator

`django-howl` comes with 3 standard operators by default:

- EqualOperator
- LowerThanOperator
- GreaterThanOperator

All operators are subclassed from `howl.operators.BaseOperator`. If you wish to add your own one, you have to subclass from the `BaseOperator` and add a compare method.

One example could be:

```
from howl.operators import BaseOperator


class MyOperator(BaseOperator):
    display_name = 'My new Operator'

    def compare(self, compare_value):
```

(continues on next page)

(continued from previous page)

```
        if compare_value < 1:
            return False

        return self.observer.value % compare_value == 0
```

Now you have to add the new operator to `settings.py`. If you want to use only your own one, then add:

`HOWL_OPERATORS = ('path.to.operator.MyOperator',)`

or if you want to use all including your one, then add:

```
HOWL_OPERATORS = (
    'howl.operators.EqualOperator',
    'howl.operators.LowerThanOperator',
    'howl.operators.GreaterThanOperator',
    'path.to.operator.MyOperator',
)
```

2.2.2 Connect to signals

Next step is, that you have to connect with the signals. There are 3 types of signals:

- alert_wait (first time the comparison failed)
- alert_notify (comparison still fails after a specific time)
- alert_clear (after failure, the comparison is working again)

Create a file named `handlers.py`. This should look like the following one:

```
from django.dispatch import receiver
from howl.models import Alert
from howl.signals import alert_clear, alert_notify, alert_wait


@receiver(alert_wait, sender=Alert)
def send_warning(sender, instance, **kwargs):
    # send info/warning notification


@receiver(alert_notify, sender=Alert)
def send_alert(sender, instance, **kwargs):
    # send critical notification


@receiver(alert_clear, sender=Alert)
def send_clear(sender, instance, **kwargs):
    # send all is working now notification
```

Last but not least you have to connect the handlers with the signals at the begining of the runtime of the wsgi process. In your `apps.py` add the following:

```
from django.apps import AppConfig


class MyAppConfig(AppConfig):
    name = 'myapp'
```

(continues on next page)

(continued from previous page)

```
    def ready(self):
        import myapp.handlers  # noqa
```

`import myapp.handlers` this is the path where you put your `handlers.py`

2.3 Next steps to be continued...

Now you can login to the admin and configure some observers and build some nice apps with it.

2.4 Changelog

2.4.1 1.1.x

1.1.0 - 2020-10-15

- add support for python 3.9
- add support for django 3.0 and 3.1
- drop support for python < 3.6
- drop support for django < 2.2

2.4.2 1.0.x

1.0.5 - 2020-03-15

- add support for python 3.7 and 3.8
- update Pipfile.lock and test environment to avoid security issues
- use github actions

1.0.4 - 2019-06-25

- increase Django version
- update Pipfile.lock and test environment to avoid security issues

1.0.3 - 2019-03-30

- add more documentation

1.0.2 - 2019-02-15

- increase test coverage
- update requirements and dev packages
- use flake8 instead of pep8 and flakes

1.0.1 - 2018-12-08

- Fix: add missing on_delete argument to migration

1.0.0 - 2018-12-07

- drop support for django < 1.11
- add support for django 2.x

2.4.3 0.1.x

0.1.13 - 2017-01-02

- refactor operator
- setting `HOWL_OPERATOR_EXTENSIONS` is renamed to `HOWL_OPERATORS`

0.1.12 - 2017-01-01

- refactor operator extension registration
- extensions can added via `HOWL_OPERATOR_EXTENSIONS` setting

0.1.11 - 2016-04-09

- added missing kwargs to get_alert_identifier method in Alert.clear

0.1.10 - 2016-04-09

- change observer value from char field to float field to avoid some compare errors

0.1.9 - 2016-04-09

- added missing migration

0.1.8 - 2016-04-09

- change observer value from integer field to char field
- extend observer get_alert_identifier method with kwargs

0.1.7 - 2016-02-11

- This change includes a slight api change in the way how Alert.set and Alert.clear is called.
- In addtion, the way how waiting_period works is improved and more clear. A waiting_period of zero means immediate critical notification without warning.

0.1.6 - 2016-01-29

- extend Alert methods with kwargs and add compare_value to signals

0.1.5 - 2016-01-01

- adjust setup file

0.1.4 - 2016-01-01

- adjust setup file

0.1.3 - 2015-12-31

- adjust setup files and add apps.py

0.1.2 - 2015-12-19

- mini bugfixes.

0.1.1 - 2015-12-16

- Moved signal logic to Alert model.

0.1.0 - 2015-12-06

- Initial release.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search