django-simple-log Documentation

Release 0.3.0

Grigory Mishchenko

Sep 24, 2020

Contents:

1 Installation 3

2 Advanced usage 5

3 Utils 7

4 Settings 9

5 Changelog 13

6 Indices and tables 15

Logging model changes on every create/update/delete (except queryset update).

CHAPTER 1

Installation

Install using pip:

```
$ pip install django-simple-log
```

Add to installed apps:

```
INSTALLED_APPS = (
    ...
    'simple_log',
    ...
)
```

Add middleware for detecting user:

```
MIDDLEWARE = [
    ...
    'request_vars.middleware.RequestVarsMiddleware',
    ...
]
```

Migrate:

```
$ python manage.py migrate
```

CHAPTER 2

Advanced usage

2.1 Manual logging

If you need to log something manually:

```
from simple_log.models import SimpleLog

SimpleLog.log(
    instance=obj,
    action_flag=SimpleLog.CHANGE,
    change_message='Some message'
)
```

2.2 Pass params to log

Add `simple_log_params` to instance:

```
class SomeView(FormView):
    ...

    def form_valid(self, form):
        form.instance.simple_log_params = {
            'user': None,
            'user_repr': 'GLaDOS',
            'change_message': 'Cake is a lie'
        }
        return super().form_valid(form)
```

2.3 Custom serializer

```
# Serializer for concrete model
class MyModel:
    ...
    simple_log_model_serializer = 'app_label.ModelName'

# Default serializer in settings
SIMPLE_LOG_MODEL_SERIALIZER = 'app_label.ModelName'
```

2.4 Custom log model

```
from simple_log.models import SimpleLogAbstractBase

from django.db import models
from django.utils.translation import ugettext_lazy as _


class ChangeLog(SimpleLogAbstractBase):
    # custom action_flag
    ERROR = 4
    ACTION_CHOICES = SimpleLogAbstractBase.ACTION_CHOICES + (
        (ERROR, 'error'),
    )

    action_flag = models.PositiveSmallIntegerField(
        _('action flag'),
        choices=ACTION_CHOICES,
        default=SimpleLogAbstractBase.CHANGE
    )

    # custom field
    user_is_staff = models.BooleanField(default=False)

    @classmethod
    def get_log_params(cls, instance, **kwargs):
        params = super(ChangeLog, cls).get_log_params(instance, **kwargs)
        user = params['user']
        if user:
            params['user_is_staff'] = user.is_staff
        return params

# in settings
SIMPLE_LOG_MODEL = 'app_label.ChangeLog'
```

CHAPTER 3

Utils

3.1 Disable logging

For temporary disable logging:

```
from simple_log.utils import disable_logging

with disable_logging():
    # create/update/delete objects
```

3.2 Disable related

For temporary disable linking related logs:

```
from simple_log.utils import disable_related

with disable_related():
    # create/update/delete objects
```

3.3 Commands

To view which models is tracking:

```
$ python manage.py view_tracking_models
```

With option `-f` you can view which fields is tracking for every model.

CHAPTER 4

Settings

4.1 SIMPLE_LOG_MODEL_LIST

Default: `()`

List of models for logging by label: 'app.Model'.

4.2 SIMPLE_LOG_EXCLUDE_MODEL_LIST

Default: `('admin.LogEntry', 'migrations.Migration', 'sessions.Session', 'contenttypes.ContentType', 'captcha.CaptchaStore')`

List of models for exclude from logging by label: 'app.Model'.

4.3 SIMPLE_LOG_EXCLUDE_FIELD_LIST

Default: `('id', 'last_login', 'password', 'created_at', 'updated_at')`

List of field names which not track.

If you need to define which fields to track for concrete model, you can add one of the properties to model: `simple_log_fields = ('id',)` or `simple_log_exclude_fields = ('password',)`.

4.4 SIMPLE_LOG_ANONYMOUS_REPR

Default: `'Anonymous'`

User representation that write to log, if anonymous user changes model.

4.5 SIMPLE_LOG_NONE_USER_REPR

Default: `'System'`

User representation that write to log, if user not detected (If middleware not working or if model changes from task or console).

4.6 SIMPLE_LOG_MODEL

Default: `'simple_log.SimpleLog'`

Model for writing logs. If you want to define your own model, you should inheritance from `simple_log.SimpleLogAbstract` and change this setting.

4.7 SIMPLE_LOG_MODEL_SERIALIZER

Default: `'simple_log.models.ModelSerializer'`

Class for serializing model fields to json.

4.8 SIMPLE_LOG_GET_CURRENT_REQUEST

Default: `'simple_log.utils.get_current_request_default'`

Function that return current request. Rewrite this setting if you already have middleware for storing current request.

4.9 SIMPLE_LOG_OLD_INSTANCE_ATTR_NAME

Default: `'_old_instance'`

Name of attribute for storing old instance of logging object.

4.10 SIMPLE_LOG_SAVE_RELATED

Default: `False`

Link related logs.

4.11 SIMPLE_LOG_SAVE_ONE_TO_MANY

Default: `False`

Track one to many fields.

4.12 SIMPLE_LOG_PROXY_CONCRETE

Default: `False`

Save content type of proxy model, instead of base.

4.13 SIMPLE_LOG_SAVE_ONLY_CHANGED

Default: `False`

Save only changed fields.

4.14 SIMPLE_LOG_DATETIME_FORMAT

Default: `None`

Format for datetime fields.

4.15 SIMPLE_LOG_DATE_FORMAT

Default: `None`

Format for date fields.

4.16 SIMPLE_LOG_TIME_FORMAT

Default: `None`

Format for time fields.

4.17 SIMPLE_LOG_FILE_NAME_ONLY

Default: `True`

Save file name only instead of path to file.

4.18 SIMPLE_LOG_EXCLUDE_RAW

Default: `False`

Don't create log entry if signal is raw (e. g. load data from fixtures).

4.19 SIMPLE_LOG_ENABLED

Default: `True`

Enable/disable log, useful for testing.

CHAPTER 5

Changelog

5.1 Version 0.3.2 (2018-11-15)

- Fix concrete content_type in admin
- Don't track unmanagement models

5.2 Version 0.3.0 (2018-08-13)

- Add request_vars library as requirements
- Add ability to use "disable_logging" and "disable_related" as decorator
- Add docs
- Drop old django versions
- Use TextField in object_repr and user_repr

CHAPTER 6

Indices and tables

- genindex
- modindex
- search