django-model-controller Documentation

Release 0.0.1

Khemanorak Khath

Apr 24, 2020

Contents

Keep tracking all of your record.

CHAPTER 1

Overview

Come to a time that you want to able to keep track on each record of data that was created or updated. This project give you the ability to answer these questions: When this record was created? When this record was updated? Who created this record? Who updated this record?

CHAPTER 2

Requirements

- Python (2.7, 3.2, 3.3, 3.4, 3.5)
- Django (1.8, 1.9, 1.10)

CHAPTER 3

Content

3.1 Getting Started

A quick start guide for getting start using Django Model Controller.

3.1.1 Setup

Install with pip and add it to your requirements:

```
$ pip install django-model-controller
```

3.1.2 Model

Usage

Extend your model with **AbstractModelController**

```
from django.db import models
from model_controller.models import AbstractModelController


class MyModel(AbstractModelController):
    name = model.CharField()
```

Now **MyModel** will include fields such as *name, created_at, updated_at, created_by* and *updated_by*.

Field Explanation

- **created_at** timestamp of when model instance created.
- **updated_at** timestamp of when model instance was updated.

- **created_by** store foreign key of User model. Recorded the user whose responsible for creating this model instance.
- **updated_by**

3.1.3 Form

In order for *created_user* and *updated_user* get record automatically, Form and View must work together. Django Model controller has already included **ModelControllerForm** for usage with form.

Usage

```
from model_controller.forms import ModelControllerForm


class MyForm(ModelControllerForm):

    class Meta:
        model = MyModel
        fields = ('name', )
```

NOTE: Since *ModelControllerForm* class is extended from *django.forms.ModelForm* the usage is the same as Django form model.

If you want to show all the fields in your model except fields from *AbstractModelController*, you can use pre-defined tuple **EXCLUDE_MODEL_CONTROLLER_FIELDS**. Example:

```
from model_controller.utils import EXCLUDE_MODEL_CONTROLLER_FIELDS
...
    class Meta:
        model = MyModel
        exclude = EXCLUDE_MODEL_CONTROLLER_FIELDS
```

3.1.4 Views

If you followed our guided all along the only think left now is View. We have already included *CreateViewMixin, UpdateViewMixin, ListViewMixin, DetailViewMixin* and *DeleteViewMixin*.

Only *CreateViewMixin* and *UpdateViewMixin* are important, other are bonus.

Each mixin is extended from Django View Generic so the usage is the same. Also, each mixin is extended from *ExtendedLoginRequiredMixin*, extended from django.braces, this mean that each view extended from our mixin is required user is authenticated (since we need to record *created_by* and *updated_by*).

Usage

Since our view is required user to login, we will need to tell view mixin what URL it should redirect to when user is not logged in.

In your settings file

```
LOGIN_URL = '/Your/Login/URL'
```

Here is the usage for view mixin.

```
from model_controller.views import CreateViewMixin, UpdateViewMixin


class MyCreateView(CreateViewMixin):
    template_name = '/template/create.html'
    model = MyModel
    form_class = MyForm
    success_url = reverse_lazy('success')


class MyUpdateView(UpdateViewMixin):
    template_name = '/template/update.html'
    model = MyModel
    form_class = MyForm
    success_url = reverse_lazy('success')
```

3.1.5 Admin

If you don't want your admin page to select the user each time it create or update you can use *ModelControllerAdmin*, which already provided for admin site usage. *ModelControllerAdmin* will automatically record current login user when an instance got created or updated.

Usage

```
from django.contrib import admin

from model_controller.admins import ModelControllerAdmin
from app.models import MyModel


class MyModelAdmin(ModelControllerAdmin):
    list_display('name', )

admin.site.register(MyModel, MyModelAdmin)
```

3.1.6 Conclusion

If there is a time that you want to keep tracking your model instance and to answer question like who create or update this, when was this create or update. For it to happen Model, Form and View must work together.

3.1.7 Thank you

Please feel free to fork and submit bug or feature request.

3.2 License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

CHAPTER 4

Indices and tables

- genindex
- modindex