
Django Crud Generator Documentation

Release 0.3.1

contraslash S.A.S.

Jun 02, 2019

Contents:

1	Getting Started	1
2	Create a blog in 5 Minutes	3
3	Indices and tables	7

CHAPTER 1

Getting Started

Django Crud Generator is not intended to create a django application from the scratch, in the opposite way just scaffold a new structure to create basic operations based on [base-django](#) package.

This package must be installed in the root of your project with the name base. To do so:

```
git clone https://github.com/contraslash/base-django base
```

If you want to create a site from scratch please check our create a blog in 5

To use [django-crud-generator](#), first install the latest version from pip

```
pip install django-crud-generator
```

Now assume you have a `Post` Model under an app called `blog`, to create the structure use

```
django-crud-generator.py --django_application_folder blog --model_name Post
```

Now you can add the `urls.py` file to your main tree.

To fully customize your templates, add `template_name` attribute to each class.

By default `django-crud-generator` assumes you already have a `base.html` template, like in all common django sites, with the default schema

```
<html>
  <head>
  </head>
  <body>
    {% block content %}

    {% endblock %}
  </body>
</html>
```


CHAPTER 2

Create a blog in 5 Minutes

We present a set of instructions in form of scripts, you can copy and paste the full section, but we recommend to execute every single command and understand what it does

```
# Create a folder for the project
mkdir myblog
# Enter the folder
cd myblog
# Create a new virtualenv
python -m venv virtualenv
# Activate the environment
source virtualenv/bin/activate
# First install django
pip install django
# Then create a project
django-admin startproject myblog .
# Y really like to have all applications in an applications folder
mkdir applications && touch applications/__init__.py
# Install django crud generator
pip install django-crud-generator
# Get a copy of base-django
git clone https://github.com/contraslash/base-django base
# enter the application folder
cd applications
# Get a base template
git clone https://github.com/contraslash/template_cdn_bootstrap
# Get an authentication module
git clone https://github.com/contraslash/authentication-django authentication
# Create your blog application
django-admin startapp blog
# Go back to root of project
cd ..
```

Now you have the shell of an app, you need to write the core.

First create your post model, in order to do so put this content on your applications/blog/models.py

```
from django.db import models
from django.contrib.auth import models as auth_models

from base import models as base_models

from . import (
    conf
)

class Post(base_models.FullSlugBaseModel):
    author = models.ForeignKey(auth_models.User, on_delete=models.CASCADE)

    url_name = conf.POST_DETAIL_URL_NAME
```

Now we create the scaffold for Post model using django-crud-generator

```
django-crud-generator.py --model_name Post --django_application_folder applications/
↳blog/ --slug
```

Now we must edit our myblog/settings.py file and at the end of the file

```
# Append the new apps to the `INSTALLED_APPS`
INSTALLED_APPS += [

    # Contraslash Apps
    'base',
    'applications.authentication',
    'applications.template_cdn_bootstrap',

    # Custom apps
    'applications.blog'
]

# And define an the LOGIN URL
LOGIN_URL = "log_in"
```

And include the configuration for the myblog/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path(
        'authentication/',
        include(
            "applications.authentication.urls",
        )
    ),
    path(
        '',
        include(
            "applications.blog.urls",
        )
    ),
]
```


Now we're ready to run the `manage.py` command.

```
# First create the new migrations
python manage.py makemigrations
# Migrate the schemas to the database
python manage.py migrate
# And create a superuser
python manage.py createsuperuser
# Now run the server
python manage.py runserver
```

Now be free to modify the template of the list and detail of your post

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`