# DjangoRestFramework Microservice Generator

***Release v0.7.1***

**Alain Ivars**

**Mar 13, 2022**

# CONTENTS:

CHAPTER

ONE

# ABOUT COOKIECUTTER-DRF-MICROSERVICE

**Cookiecutter-drf-microservice is a ready-to-use API skeleton generator:**

- generates it,

**And you:**

- add your unittest and endpoints,

**And it will help you to:**

- generate the documentation with Coreapi,
- test it with Tox,
- package it (TODO with) Docker,
- deploy it (TODO with) Terraform or Ansible

It sounds simple and it is. Take a look at Drf-microservice it's now generated by Cookiescutter-drf-microservice.

Something disturb you in the code? Don't hesitate to open a an issue and contribute.

CHAPTER

TWO

# RELEASES NOTES

- 0.7.1: Remove all .md file, update the doc, the docker config file
- 0.7.0: cookiecutter-drf-microservice got it own separate repository
- 0.6.1: Update dependencies
- 0.6.0: total refactoring for add cookiecutter
- 0.5.2: fix dependencies security alert
- 0.5.1: fix some document presentation on github and pypi
- 0.5.0: Initial publication version

CHAPTER

THREE

# BUGS AND EVOLUTION POLICIES

When you will find a bug or propose an evolution create a ticket on:

- Issue Cookiescutter-drf-microservice if it's about the generation process
- Issue Drf-microservice if it's about a functionality in the generated drf process

## 3.1 Usage

- If needed install https://github.com/audreyr/cookiecutter or

```
pip install cookiecutter
```

- Cookiescutter will generate it for you

```
cookiecutter gh:alainivars/cookiecutter-drf-microservice
↪
↪          00:31:00
github_username [my-github-user-name]: alainivars
github_repository_name [my-repository]: drf-microservice
app_name [my_app]: my_api
email [my-email@my-domain.my]: alainivars@gmail.com
description [The description of my drf app]: A simple demo on how to use
↪cookiecutter-drf-microservice generator
```

For all operation with the new "my-drf-microservice" I invite you to go at Drf-microservice

- **Now we just jump in the new directory and run tox to ::**
  - be sure that everything as worked fine
  - generate the documentation
  - generate an virtualenv

```
cd drf-microservice
tox
```

- An virtualenv is already ready for you at

```
tox -l
py36-django222
```

- or you can create your

```
python3 -m venv /pass/to/venv
```

- for bash, zsh

```
source .tox/py36-django222/bin/activate
```

- for fish

```
source .tox/py36-django222/bin/activate.fish
```

- for bash, zsh

```
SECRET_KEY=my_secret_key python manage.py makemigrations
SECRET_KEY=my_secret_key python manage.py migrate
SECRET_KEY=my_secret_key python manage.py createsuperuser
```

- for fish

```
env SECRET_KEY=my_secret_key python manage.py makemigrations
env SECRET_KEY=my_secret_key python manage.py migrate
env SECRET_KEY=my_secret_key python manage.py createsuperuser
```

- then run it

```
SECRET_KEY=my_secret_key python manage.py runserver
```

- if you have any problem or you want enable the debug mode

```
ENABLE_DEBUG=1
```

## 3.2 Functionalities

- support basic auth
- support token auth
- endpoint json file POST,GET
- endpoint login/logout
- endpoint get tocken
- postgreSQL support
- doc modular & less duplicated

### 3.2.1 Todo

- AWS ssm secret
- endpoint json file DELETE,PUT?
- **create different version:**
  - Aws S3 support (in progress)
  - Aws RDS support

  - Aws Elastisearch support
  - Redis support
  - Aerospike support
  - ...

## 3.3 DevOps tools

- the docker-compose configuration file
- endpoint get status Nagios/Icinga2

### 3.3.1 Todo

- the docker-image configuration file (in progress)
- the Packer configuration file (in progress)
- the Terraform configuration file AWS (in progress)
- the Terraform configuration file GCD
- the Terraform configuration file Azure
- add getSentry support
- add Aws Cloudwatch support
- the Ansible configuration file AWS
- the Ansible configuration file GCD
- the Ansible configuration file Azure
- the Juju configuration file AWS
- the Juju configuration file GCD
- the Juju configuration file Azure
- Make static doc more modular & less duplicated

## 3.4 Interact with the API

To see the documentation for the API In development mode, login at

```
curl --request POST \
  --url http://127.0.0.1:8000/api-auth/login/ \
  --header 'content-type: application/json' \
  --data '{
    "username": "admin",
    "password": "admin"
    }'
```

Actually the default mode is "development" (same to the state of this project) in that mode a default login is the the db with username='admin' password='admin' you will get back in return your token:

```
{"key":"400a4e55c729ec899c9f6ac07818f2f21e3b4143"}
```

Then open to see the full auto-generated documentation of you API:

```
curl --request GET \
  --url http://127.0.0.1:8000/docs/ \
  --header 'authorization: Basic YWRtaW46YWRtaW4='
```

or by if BasicAuthentication is disabled and that wil be normally the case in prod and QA we use the Token:

```
curl --request GET \
  --url http://127.0.0.1:8000/docs/ \
  --header 'authorization: Token 400a4e55c729ec899c9f6ac07818f2f21e3b4143'
```

Then open

```
http://127.0.0.1:8000/docs/
```

../media/docs.png

## 3.5 Testing

You can run the tests by

```
SECRET_KEY=my_secret_key python manage.py test
```

or by

```
python setup.py test
```

or by

```
DJANGO_SETTINGS_MODULE={{cookiecutter.app_name}}.config.local SECRET_KEY=my_secret_
↪key pytest
```

## 3.6 Security check

Before dockerization for deployment to production, don't forget to check if by

```
SECRET_KEY=my_secret_key python manage.py check --deploy
```

## 3.7 Build and run the image with Docker

Build and run with docker-compose:

```
docker-compose up
```

Then login, see API documentation

**Warning:** WORK IN PROGRESS, not existing actually

Build the Docker image:

```
docker build -t my-drf -f Dockerfile.drf-microservice .
docker build -t my-nginx -f Dockerfile.nginx .
```

Run the container:

```
docker network create my-network
docker run -d --name drf --net my-network -v /app my-drf
docker run -d --name nginx --net my-network -p "5000:80" my-nginx
```

If you want to change the port binding, it's here...

## 3.8 If you Use Aws

Aws secret required ???:: WORK IN PROGESS

APPNAME_USERNAME_PASSWD => a client API password SECRET_KEY => the secret key

Aws Env required:

```
AWS_REGION_NAME => default="eu-east-1"
AWS_APPNAME_SECRET_NAME =>The name of the secret bucket
```

## 3.9 Releases Notes

- 0.7.1: doc modular & less duplicated, the docker config file

- 0.7.0: cookiecutter-drf-microservice got it own separate repository
- 0.6.1: Update dependencies
- 0.6.0: total refactoring for add cookiecutter
- 0.5.2: fix dependencies security alert
- 0.5.1: fix some document presentation on github and pypi
- 0.5.0: Initial publication version

## 3.10 Indices and tables

- genindex
- modindex
- search