

Deploy Manual (Windows)

(V-2.0)

(All copyrights reserved)

Project: ucdgrapenews.com

GitHub: <https://github.com/ucd-nlmsc-teamproject/Gungnir-Repo.git>

Date: 2017-08-20

Authors: Shen Wenrui

Email: Wenrui.shen@ucdconnect.ie

1. Introduction

This manual is for the demo deployment on Windows platform, it does not support full functions, such as celery auto scheduler, in this Manual. This deployment is used for developing and testing.

2. Environment Requirements:

- 2.1. Operating system: Windows 10;
- 2.2. Commends environment: DOS;
- 2.3. Python 3;
- 2.4. Django 1.10 (or above);
- 2.5. PostgreSQL;
- 2.6. Install through requirements.txt:
`pip install -r requirements.txt`
- 2.7. Install manually:
`pip3 install --upgrade pip3`
`pip3 install django`
`pip3 install django-rest-framework`
`pip3 install psycopg2`
`pip3 install newspaper`
`pip3 install arrow`
`pip3 install simhash`

3. Install and connect PostgreSQL:

- 3.1. Related codes:
 - - dataCollector/models.py:
 - - model_interfaces.py (up layer interfaces for applications);
 - - model_test.py (including examples);
- 3.2. Create Postgresql Database:
Please add PATH ahead.
CMD:> `psql -U postgres -W`
`# CREATE USER djangouser01;`
`# CREATE DATABASE djangodb01 OWNER djangouser01;`

```
username: postgres (superuser)
          djangouser01
password: presqlswr123
port:    5432
address: 127.0.0.1
```

3.3. Connect PostgreSQL:

Change gungnir/settings.py:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'djangodb01',
        'USER': 'postgres',
        'PASSWORD': 'presqlswr123',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    }
}
```

3.4. Migrate Postgresql Database:

3.4.1. Run cmd to create migrations for those changes:

```
python manage.py makemigrations
```

3.4.2. Run cmd to apply those changes to the database:

```
python manage.py migrate
```

(Optional): If you want to see SQL Statements:

```
python manage.py sqlmigrate dataCollector 0001
```

3.5. Create Super User:

```
python manage.py createsuperuser
username: gungnir_admin
email: wenrui.shen@ucdconnect.ie
ps: gapw2017
```

4. Install and set Celery:

(For details, please refer 'Data_Collector_Design')

4.1. Install RabbitMQ and Erlang:

```
Download RabbitMQ and Erlang and install them;
RabbitMQ start;
```

4.2. Install Celery:

```
pip install django-celery
```

4.3. Running and testing Celery:

```
Python manage.py migrate
```

```
Start RabbitMQ;
```

(mac):

```
brew services restart rabbitmq
brew services start rabbitmq
brew services stop rabbitmq
```

Start celery worker:

Celery -A gungnir worker --loglevel=info

Start celery Beat:

Celery beat -A gungnir

Testing:

```
from dataCollector.tasks import get_topicsv2
insert_topic.delay()
get_topicsv2()
```

5. Install and build frond-end:

- 5.1. Install node.js (Only once);
- 5.2. Install npm (Only once);
- 5.3. Install requirements of front-end:
npm install;
- 5.4. Build front-end:
npm run build

6. Initialized data collection:

7. Check supported domains:

gungnir/Settings.py:

```
DEBUG = False      # False in remote env.
ALLOWED_HOSTS = ['127.0.0.1',
                  'localhost ',
                  '.csi6220-2-vm2.ucd.ie',
                  'ucdgrapenews.com',
                  'www.ucdgrapenew.com']
```

8. Start demo testing:

- 8.1. Clean the database:
python manage.py flush
- 8.2. Set independent Django running environment:
(Only for Windows)
set DJANGO_SETTINGS_MODULE=gungnir.settings
- 8.3. Run commands:
python manage.py shell
- 8.4. Initialized data collection:
python Init_data_collector.py
- 8.5. Run Django server:
python manage.py runserver