



00. 배포 구조

01. 배포 사전 준비

- 1.1 배포 사전 체크 사항
- 1.3 github 레퍼지토리 생성
- 1.4 django-heroku

02. 배포를 위한 설정

- 2.1. Procfile
- 2.2. Gunicorn
- 2.3. runtime.txt
- 2.4. requirements.txt

03. 배포

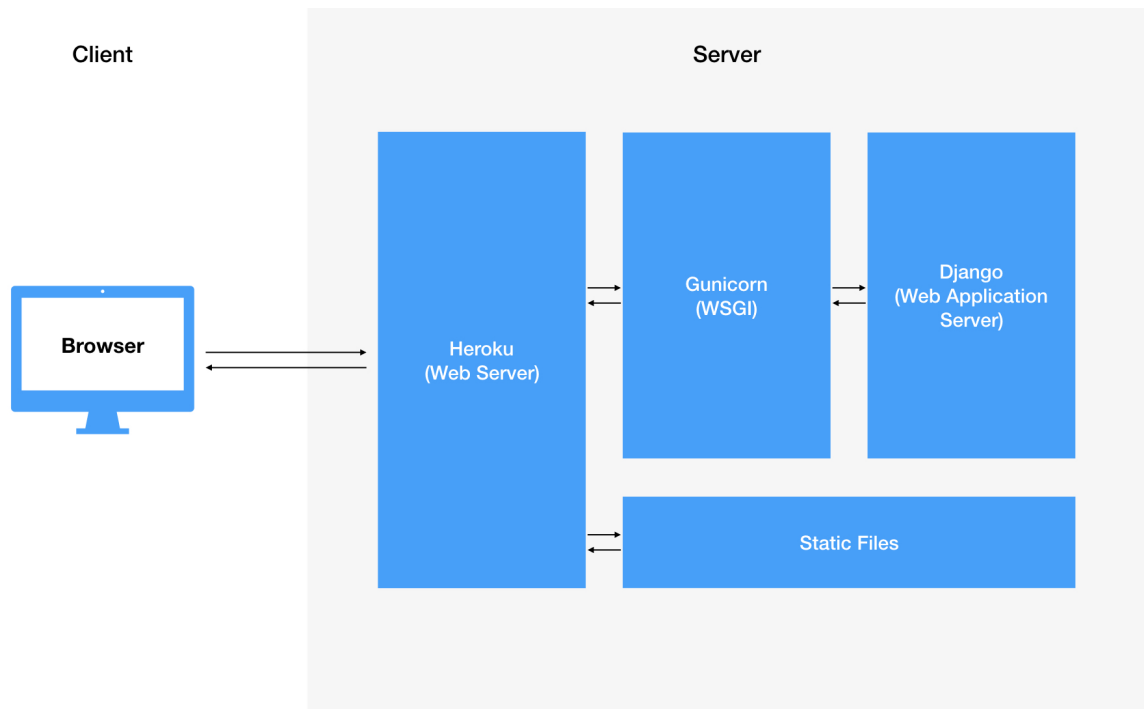
- 3.1. Heroku 회원가입 및 CLI 에서 로그인
- 3.2. Heroku 에 Django App 배포
- 3.3. 배포한 Django App Initializing
- 3.4. Social Login (Optional)

04. References



문서 저작권은 주식회사 해피해킹에 있으며 무단 배포 및 수정을 금합니다.

00. 배포 구조



- Gunicorn 을 세팅한 django 프로젝트를 heroku 에 올려서 배포한다.
- Heroku 에 배포 시 Web Server 를 자동으로 구축해주기 때문에 WSGI 및 Web Application Server 만 세팅한다.

01. 배포 사전 준비

[배포 전 체크리스트 참고자료](#)

1.1 배포 사전 체크 사항

```
# master 브랜치에서 배포 브랜치 생성
$ git checkout -b feature/deploy
```

```
# base.py
# 기존 개발시 사용하던 것들
STATIC_URL = '/static/'

STATICFILES_DIRS = [
    ...
]

# 배포 때 추가
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

1.3 github 레퍼지토리 생성

`.gitignore` 해당 코드 추가

- [gitignore.io](https://github.com/gitignore-io) 에서 **Python, Django, venv** 검색 후 추가하기

```
$ touch .gitignore
```

```
# Text backup files
*.bak

# Database
*.sqlite3

# 환경 설정 관련 파일(뒤에서 설정할 것)
.env

# venv 파일
venv/

# vscode 관련 세팅 파일
.vscode/
```

원격 저장소에 업로드

- Heroku는 git 형상 관리 시스템과 밀접하게 통합 되어있다.
- git을 이용하여 활성화된 시스템에 수정 사항의 업로드 및 동기화를 수행한다.
- **gitlab에 repo를 만들어서 진행하자!**

```
# 새로운 폴더에서 배포 진행하는 경우

$ git commit -m "deploy commit"
$ git push origin master
```

1.4 django-heroku

▼ [django-heroku?](#)



django-heroku

django-heroku automatically configures our Django application to work on Heroku.

This will automatically configure `DATABASE_URL`, `ALLOWED_HOSTS`, `WhiteNoise` (for static assets), `Logging`, and `Heroku CI` for your application.

If you set the `SECRET_KEY` environment variable, it will automatically be used in your Django settings, too!

- Django 앱이 heroku 에 동작할 수 있도록 자동으로 환경을 설정해주는 라이브러리.
- Heroku에서 만든 라이브러리.
- Django를 heroku 서버에 배포하기 위해 사용한다.

```
$ pip install django-heroku
```

```
# settings.py 최하단 (settings 가 나뉘어져 있다면 base.py + production.py 최하단에 두어야 함)
...

# base.py + production.py 둘 최하단

# Configure Django App for Heroku.
import django_heroku
django_heroku.settings(locals())
```

▼ PostgreSQL (Mac OS)

- Mac 의 경우 PostgreSQL 설치 필요. (윈도우는 설치 필요 없음)
- Heroku에서 DB를 postgresql 를 쓰게 강제되어 있다.

```
$ brew install postgresql
```

02. 배포를 위한 설정

- Django에서 배포를 위한 설정과 관련된 파일
 - **runtime.txt**: 프로그래밍 언어와 사용 버전.
 - **requirements.txt**: Django를 포함한 파이썬 관련 라이브러리 의존성.
 - **Procfile**: 웹 어플리케이션을 구동하기 위해 실행되어야 하는 프로세스의 목록. 장고의 예를 들자면, Gunicorn 웹 어플리케이션 서버(.wsgi 스크립트와 함께) 가 될것이다.
 - **wsgi.py**: Heroku 환경에서 Django 어플리케이션을 호출 하기 위한 **WSGI** 설정.

2.1. Procfile

- 웹 어플리케이션을 구동하기 위해 **실행되어야 하는 프로세스의 목록이 정의**되는 곳이다.
- 지금 배포하는 **django 프로젝트**에서는 **gunicorn 웹 어플리케이션 서버**가 될것이다.
- (주의) Django 기준으로 프로젝트 이름은 `$ python manage.py startproject 프로젝트_이름` 으로 만든 프로젝트를 의미한다.

```
accounts
articles
myform(프로젝트)
venv/
.env
db.sqlite3
.gitignore
manage.py
Procfile
requirements.txt
```

- `articles` , `accounts` 가 app 이라면 `myform` 이 프로젝트가 된다.

```
$ touch Procfile
```

```
# 프로젝트폴더/Procfile
```

```
web: gunicorn [프로젝트_이름].wsgi --log-file -
```



"web:" 구문은 Heroku에게 이것이 웹 다이노이며 HTTP 트래픽을 받을수 있다는 것을 알려 준다.

다이노(Dyno)는 고립적이고 가상화된 Unix 컨테이너이며 어플리케이션을 실행하는 데 필요한 환경 제공.

이 다이노에서 시작할 프로세스는 **gunicorn(WSGI)** 인데 Heroku 가 추천하는 인기있는 웹 어플리케이션 서버이다.

[프로젝트_이름].wsgi (어플리케이션 뼈대로 생성된: `/프로젝트_이름/wsgi.py`) 모듈의 설정 정보를 이용하여 Gunicorn 을 구동시킨다.

2.2. Gunicorn

```
$ pip install gunicorn
```

2.3. runtime.txt

- 현재 프로젝트가 사용 중인 python 버전을 입력한다.
- `python -V` 명령어로 학생들이 사용하는 파이썬 버전 미리 확인한다.

```
$ python -V
Python 3.7.3
```

```
runtime.txt
```

```
$ touch runtime.txt
```

```
# 프로젝트폴더/runtime.txt
```

```
python-3.7.3
```

2.4. requirements.txt

- Django 를 포함한 파이썬 관련 라이브러리 의존성.
- Heroku 는 환경을 재구성할 때 `requirements.txt` 에 패키지들을 자동적으로 설치한다.

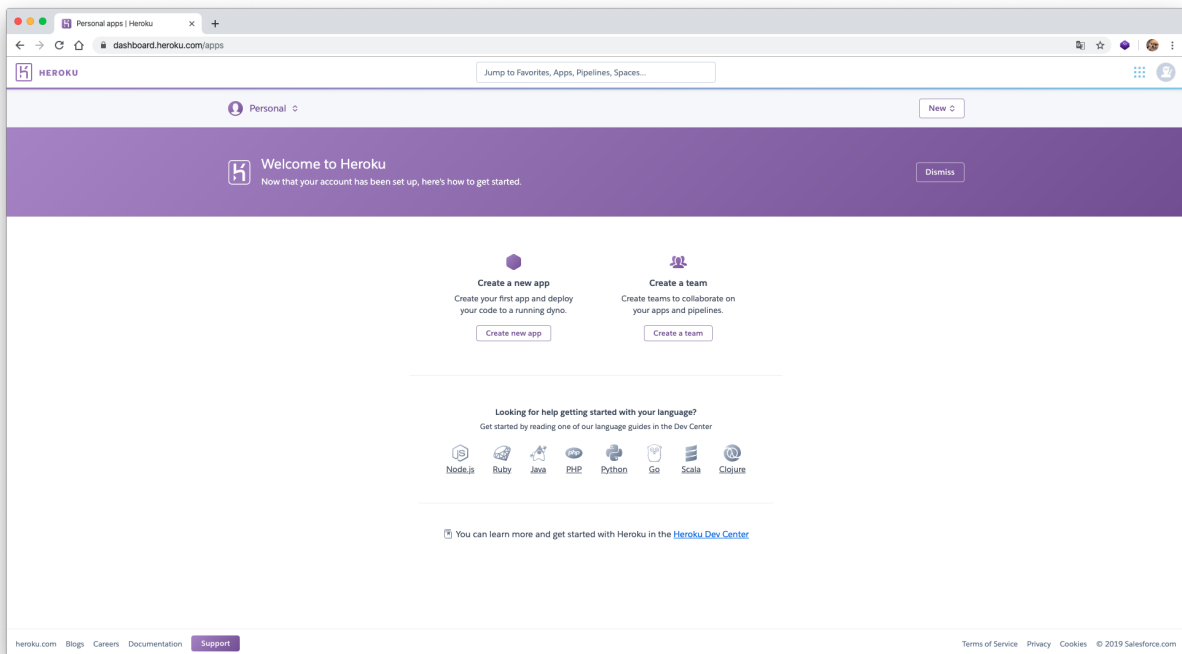
```
$ pip freeze > requirements.txt
```

03. 배포

3.1. Heroku 회원가입 및 CLI 에서 로그인

헤로쿠 회원가입

- 회원가입이 완료되면 아래와 같은 dashboard가 나타난다.

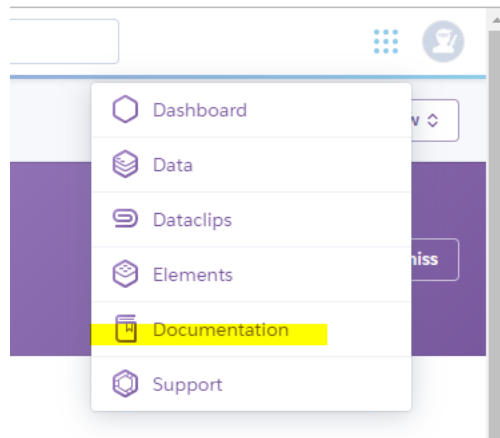


Heroku cli 윈도우 64 bit 버전 설치

▼ 참고 사이트

<https://devcenter.heroku.com/articles/heroku-cli>

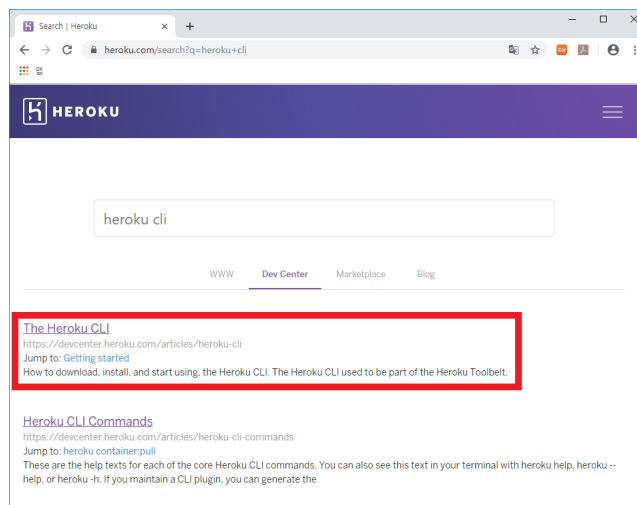
- Heroku 에서 heroku cli 검색
 - 혹은 아래 사진의 버튼에서 Documentation 클릭 후 **The Heroku CLI** 를 찾아 들어가도 된다.



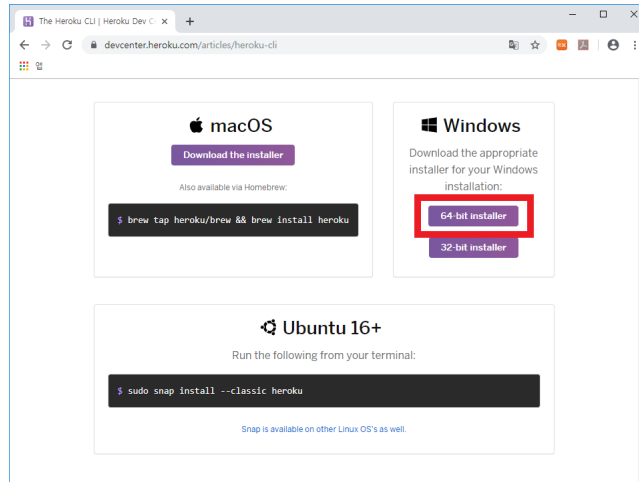
Heroku Essentials

⚡ [Get Started](#)

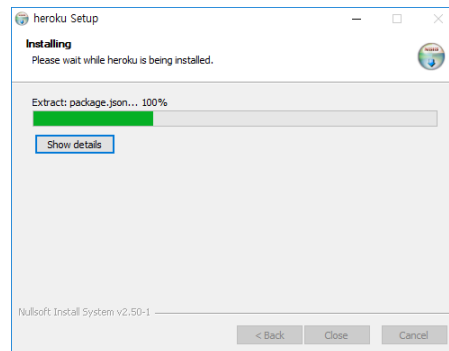
📖 [The Heroku CLI](#)



- 64bit 의 윈도우 버전을 다운로드한다.



- 전부 "다음" 을 눌러 설치한다.



- 설치가 완료되면 VS Code 를 완전히 종료 후 다시 시작!

- ▼ windows 용 설치가 안될 경우

```
$ npm install -g heroku
```

heroku 명령어로 사용할 수 있는 명령어 목록 보기

```
$ heroku
```



```
MINGW64/c/Users/student
student@USER MINGW64 ~
$ heroku
CLI to interact with Heroku

VERSION
heroku/7.33.3 win32-x64 node-v11.14.0

USAGE
$ heroku [COMMAND]

COMMANDS
access      manage user access to apps
addons      tools and services for developing, extending, and operating
            your app
apps        manage apps on Heroku
auth        check 2fa status
authorizations OAuth authorizations
autocomplete display autocomplete installation instructions
base
buildpacks  scripts used to compile apps
certs       a topic for the ssl plugin
ci          run an application test suite on Heroku
clients     OAuth clients on the platform
config      environment variables of apps
```

cli 로 heroku 에 로그인

```
$ heroku login
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

/heroku master ↑1 $ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/d6a3d18d-fe69-452d-9cbf-477211a54401
heroku: Waiting for login... ::
```

- **q** 를 제외한 아무 키나 누르면 브라우저에서 로그인을 할 수 있게 된다.

```
/heroku master ↑1 $ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/browser/d6a3d18d-fe69-452d-9cbf-477211a54401
Logging in... done
Logged in as [redacted]@gmail.com
/heroku master ↑1 $
```

3.2. Heroku 에 Django App 배포

Heroku app 생성

- app_name 을 입력하지 않을 경우 랜덤으로 만들어 준다.

```
$ heroku create [app_name]
```

```
/heroku master ↑1 $ heroku create sspy2-2-test-django-deployment
Creating ● sspy2-2-test-django-deployment... done
https://sspy2-2-test-django-deployment.herokuapp.com/ | https://git.heroku.com/sspy2-2-test-django-deployment.git
/heroku master ↑1 $
```

- `git remote -v` 해보면 Heroku remote 등록이 된 것을 확인할 수 있다.

```
/heroku master ↑1 $ git remote -v
heroku https://git.heroku.com/sspy2-2-test-django-deployment.git (fetch)
heroku https://git.heroku.com/sspy2-2-test-django-deployment.git (push)
origin https://github.com/sspy2-2/crud.git (fetch)
origin https://github.com/sspy2-2/crud.git (push)
/heroku master ↑1 $
```

Heroku 에 환경변수 등록

`.env` 파일은 git 버전관리에서 제외시켰기 때문에 환경변수를 heroku 에 따로 등록한다.

```
#1. .env 파일에서 복사해온다.
$ heroku config:set SECRET_KEY='...' # 반드시 single로 작성

#2.
$ heroku config:set DEBUG=False
```

```
/heroku master ↑1 $ heroku config:set SECRET_KEY='bw_@dk5rxv4p*3%$9ild2#w&)-o04@*c38)1ggvv=_2x+sd$#@'
Setting SECRET_KEY and restarting ● sspy2-2-test-django-deployment... done, v3
SECRET_KEY: bw_@dk5rxv4p*3%$9ild2#w&)-o04@*c38)1ggvv=_2x+sd$#@
/heroku master ↑1 $ heroku config:set DEBUG=True
Setting DEBUG and restarting ● sspy2-2-test-django-deployment... done, v4
DEBUG: True
/heroku master ↑1 $
```

- ▼ `heroku config:set` 이 동작하지 않을 경우 Heroku app에서 settings에서 직접 등록한다.

Personal > sspy2-2-test-django-deployment

Overview Resources Deploy Metrics Activity Access Settings

Name sspy2-2-test-django-deployment Edit

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Reveal Config Vars

Personal > sspy2-2-test-django-deployment ★ Open app More

Overview Resources Deploy Metrics Activity Access **Settings**

Name sspy2-2-test-django-deployment Edit

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some addons come with their own.

Config Vars Hide Config Vars

DATABASE_URL	postgres://asdtmvtbimbqmw:00fa4eff870e	✎ ✖
DEBUG	False	✎ ✖
SECRET_KEY	bw_@dk5rxv4p*3%\$9i\ld2#w&)-o04@*c38)1g	✎ ✖
KEY	VALUE	Add

- Key : SECRET_KEY, DEBUG
- Value : 을 제외한 Key 값, False

Heroku 에 프로젝트 push

- commit

```
$ git add .
$ git commit -m "commit message"
# commit 전에 지금까지 오타 없이 작성했는지 반드시 확인해야 한다.
```

```
$ git push heroku 브랜치이름:master # requirements.txt 의존성 패키지를 설치하고 서버를 구성한다.
$ heroku config # 설정 정보 확인
```

- `Error while running '$ python manage.py collectstatic --noinput'.` 이라는 에러가 발생하면 에러 메시지에 나와있는대로 아래의 명령어를 수행하고 다시 push를 진행한다.

```
$ heroku config:set DISABLE_COLLECTSTATIC=1
$ git push heroku feature/deploy:master
```

```

remote: ----> Compressing...
remote: Done: 66.1M
remote: ----> Launching...
remote: Released v7
remote: https://sspy2-2-test-django-deployment.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sspy2-2-test-django-deployment.git
 * [new branch] master -> master
/heroku master +1 $ heroku config
=== sspy2-2-test-django-deployment Config Vars
DATABASE_URL: postgres://asdtmvtbimbqmw:00fa4eff870ea4088207fb04adad0b3b2b4aee00c533cc4f2381491350c7fab5@ec2-23-21-94-99.compute-1.amazonaws.com:5432/dc477c68b6vtd1
DEBUG: True
SECRET_KEY: bw_@dk5rxv4p*3%$9ild2#wk)-o84@*c38)lggvv=_2x+sd$j@
/heroku master +1 $

```

▼ 에러 발생 상황

3.3. 배포한 Django App Initializing

Migrate

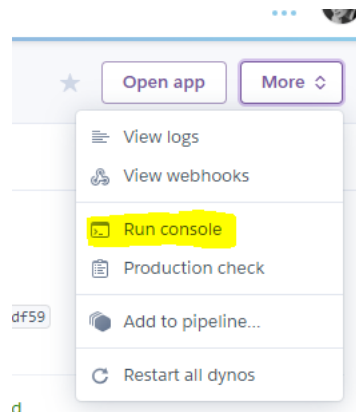
▼ ETIMEDOUT 에러

```

/heroku master +1 $ heroku run python manage.py migrate
Running python manage.py migrate on ● sspy2-2-test-django-deployment... !
 * ETIMEDOUT: connect ETIMEDOUT 50.19.103.36:5000

```

- 5000 port 가 막혀있어서 발생
- `heroku run` 으로 시작하는 모든 명령어들을 헤로쿠 웹 콘솔에서 직접 진행해야 한다.



```

# $ heroku run python manage.py makemigrations
$ heroku run python manage.py migrate

```

```

/heroku master ↑1 $ heroku run python manage.py migrate
Running python manage.py migrate on ● sspy2-2-test-django-deployment... up, run.2686 (Free)
Operations to perform:
  Apply all migrations: admin, articles, auth, contenttypes, jobs, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying articles.0001_initial... OK
  Applying articles.0002_comment... OK
  Applying articles.0003_auto_20190925_0926... OK
  Applying articles.0004_article_image... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying jobs.0001_initial... OK
  Applying jobs.0002_job_profile_image... OK
  Applying sessions.0001_initial... OK

```

Data Seeding (Optional)

- Data seeding 이 필요한 경우에만 진행

```
$ heroku run python manage.py loaddata example.json
```

Admin

```
$ heroku run python manage.py createsuperuser
```

```

/heroku master ↑1 $ heroku run python manage.py createsuperuser
Running python manage.py createsuperuser on ● sspy2-2-test-django-deployment... up, run.6049 (Free)
사용자 이름 (leave blank to use 'u39888'): jason
이메일 주소 :
Password:
Password (again):
Superuser created successfully.
/heroku master ↑1 $

```

▼ table이 없다는 에러가 발생하는 경우

- migrations 폴더 내 파일 삭제 및 db.sqlite3 파일 삭제

```

$ git add .
$ git commit -m "deleted ..."
$ git push heroku feature/deploy:master

```

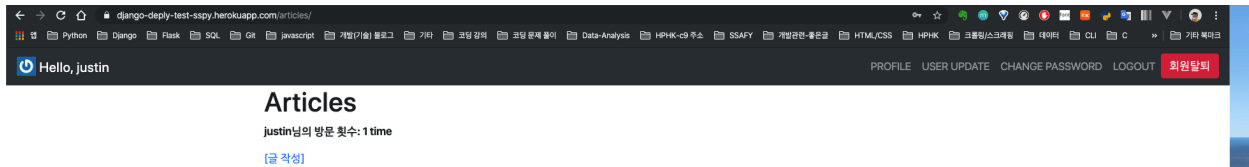
```

$ heroku run python manage.py makemigrations # 여기부터 다시 진행
$ heroku run python manage.py migrate
$ heroku run python manage.py createsuperuser

```

배포 확인

```
$ heroku open
```



수정 사항 발생 시

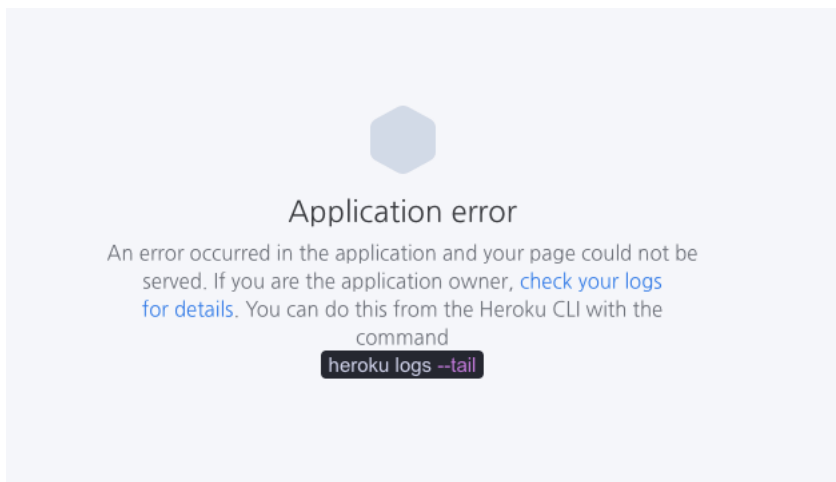
- 에러 혹은 수정 사항이 생겨 변경을 하면 반드시 아래와 같이 **add, commit 이후에** 다시 heroku 서버로 **push**를 진행하고 **heroku open** 을 해야 한다.
- 에러 해결 후에 아래와 같은 절차 수행

```
# 배포에 실패하고 수정 사항이 있으면 수정 이후에 반드시 add, commit 진행

$ git add .
$ git commit -m "commit message"
$ git push heroku 브랜치명:master
$ heroku open
```

에러 발생 시

- 아래와 같은 에러가 발생했다고 가정해보면 **heroku logs --tail** 으로 에러를 찾아주면 된다.



▼ 에러 메시지

- 모듈이 없다는 걸 확인하고 그에 맞게 수정을 진행한다.
- 이후에 **add** → **commit** → **push** 의 과정을 거치고 다시 **heroku open**

```
(venv) → django_form git:(master) heroku logs --tail
2019-11-13T00:05:50.714667+00:00 app[web.1]: self.callable = self.load()
2019-11-13T00:05:50.714668+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 49, in load
2019-11-13T00:05:50.714670+00:00 app[web.1]: return self.load_wsgiapp()
2019-11-13T00:05:50.714671+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/si
```

```

te-packages/gunicorn/app/wsgiapp.py", line 39, in load_wsgiapp
2019-11-13T00:05:50.714673+00:00 app[web.1]: return util.import_app(self.app_uri)
2019-11-13T00:05:50.714674+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/util.py", line 331, in import_app
2019-11-13T00:05:50.714676+00:00 app[web.1]: mod = importlib.import_module(module)
2019-11-13T00:05:50.714686+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/importlib/__init__.py", line 127, in import_module
2019-11-13T00:05:50.714679+00:00 app[web.1]: return _bootstrap._gcd_import(name[level:], package, level)
2019-11-13T00:05:50.714680+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:05:50.714682+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:05:50.714683+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 953, in _find_and_load_unlocked
2019-11-13T00:05:50.714685+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
2019-11-13T00:05:50.714686+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:05:50.714688+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:05:50.714689+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 965, in _find_and_load_unlocked
2019-11-13T00:05:50.714694+00:00 app[web.1]: ModuleNotFoundError: No module named 'django_form'
2019-11-13T00:05:50.714902+00:00 app[web.1]: [2019-11-13 00:05:50 +0000] [11] [INFO] Worker exiting (pid: 11)
2019-11-13T00:05:50.760562+00:00 app[web.1]: [2019-11-13 00:05:50 +0000] [4] [INFO] Shutting down: Master
2019-11-13T00:05:50.760657+00:00 app[web.1]: [2019-11-13 00:05:50 +0000] [4] [INFO] Reason: Worker failed to boot.
2019-11-13T00:05:50.828009+00:00 heroku[web.1]: Process exited with status 3
2019-11-13T00:05:56.836509+00:00 heroku[web.1]: Starting process with command `gunicorn django_form.wsgi --log-file -`
2019-11-13T00:05:59.108217+00:00 heroku[web.1]: State changed from starting to crashed
2019-11-13T00:05:58.991197+00:00 app[web.1]: [2019-11-13 00:05:58 +0000] [4] [INFO] Starting gunicorn 20.0.0
2019-11-13T00:05:58.991725+00:00 app[web.1]: [2019-11-13 00:05:58 +0000] [4] [INFO] Listening at: http://0.0.0.0:28336 (4)
2019-11-13T00:05:58.991825+00:00 app[web.1]: [2019-11-13 00:05:58 +0000] [4] [INFO] Using worker: sync
2019-11-13T00:05:58.995855+00:00 app[web.1]: [2019-11-13 00:05:58 +0000] [11] [INFO] Booting worker with pid: 11
2019-11-13T00:05:59.001229+00:00 app[web.1]: [2019-11-13 00:05:58 +0000] [11] [ERROR] Exception in worker process
2019-11-13T00:05:59.001232+00:00 app[web.1]: Traceback (most recent call last):
2019-11-13T00:05:59.001251+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/arbiter.py", line 58
2019-11-13T00:05:59.001253+00:00 app[web.1]: worker.init_process()
2019-11-13T00:05:59.001255+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/workers/base.py", li
2019-11-13T00:05:59.001256+00:00 app[web.1]: self.load_wsgi()
2019-11-13T00:05:59.001257+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/workers/base.py", li
2019-11-13T00:05:59.001259+00:00 app[web.1]: self.wsgi = self.app.wsgi()
2019-11-13T00:05:59.001261+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/base.py", line 6
2019-11-13T00:05:59.001262+00:00 app[web.1]: self.callable = self.load()
2019-11-13T00:05:59.001263+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", li
2019-11-13T00:05:59.001265+00:00 app[web.1]: return self.load_wsgiapp()
2019-11-13T00:05:59.001266+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", li
2019-11-13T00:05:59.001268+00:00 app[web.1]: return util.import_app(self.app_uri)
2019-11-13T00:05:59.001269+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/util.py", line 331,
2019-11-13T00:05:59.001271+00:00 app[web.1]: mod = importlib.import_module(module)
2019-11-13T00:05:59.001272+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/importlib/__init__.py", line 127, in import
2019-11-13T00:05:59.001274+00:00 app[web.1]: return _bootstrap._gcd_import(name[level:], package, level)
2019-11-13T00:05:59.001275+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:05:59.001276+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:05:59.001278+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 953, in _find_and_load_unlocked
2019-11-13T00:05:59.001279+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
2019-11-13T00:05:59.001281+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:05:59.001282+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:05:59.001283+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 965, in _find_and_load_unlocked
2019-11-13T00:05:59.001285+00:00 app[web.1]: ModuleNotFoundError: No module named 'django_form'
2019-11-13T00:05:59.001354+00:00 app[web.1]: [2019-11-13 00:05:59 +0000] [11] [INFO] Worker exiting (pid: 11)
2019-11-13T00:05:59.028122+00:00 app[web.1]: [2019-11-13 00:05:59 +0000] [4] [INFO] Shutting down: Master
2019-11-13T00:05:59.028228+00:00 app[web.1]: [2019-11-13 00:05:59 +0000] [4] [INFO] Reason: Worker failed to boot.
2019-11-13T00:05:59.088171+00:00 heroku[web.1]: Process exited with status 3
2019-11-13T00:07:57.527370+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=django-deploy-test
2019-11-13T00:07:58.954141+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/favicon.ico" host=django
2019-11-13T00:08:19.815968+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=django-deploy-test
2019-11-13T00:08:21.193986+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/favicon.ico" host=django
2019-11-13T00:11:13.488842+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=django-deploy-test
2019-11-13T00:11:14.923453+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/favicon.ico" host=django
2019-11-13T00:36:29.256856+00:00 heroku[web.1]: State changed from crashed to starting
2019-11-13T00:36:35.313584+00:00 heroku[web.1]: Starting process with command `gunicorn django_form.wsgi --log-file -`
2019-11-13T00:36:37.856014+00:00 heroku[web.1]: State changed from starting to crashed
2019-11-13T00:36:37.708564+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [4] [INFO] Starting gunicorn 20.0.0
2019-11-13T00:36:37.713734+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [4] [INFO] Listening at: http://0.0.0.0:28811 (4)
2019-11-13T00:36:37.714059+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [4] [INFO] Using worker: sync
2019-11-13T00:36:37.722202+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [11] [INFO] Booting worker with pid: 11
2019-11-13T00:36:37.729268+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [11] [ERROR] Exception in worker process
2019-11-13T00:36:37.729272+00:00 app[web.1]: Traceback (most recent call last):
2019-11-13T00:36:37.729274+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/arbiter.py", line 58
2019-11-13T00:36:37.729277+00:00 app[web.1]: worker.init_process()
2019-11-13T00:36:37.729279+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/workers/base.py", li
2019-11-13T00:36:37.729280+00:00 app[web.1]: self.load_wsgi()
2019-11-13T00:36:37.729282+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/workers/base.py", li
2019-11-13T00:36:37.729284+00:00 app[web.1]: self.wsgi = self.app.wsgi()
2019-11-13T00:36:37.729286+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/base.py", line 6
2019-11-13T00:36:37.729289+00:00 app[web.1]: self.callable = self.load()

```

```

2019-11-13T00:36:37.729291+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 11, in <module>
2019-11-13T00:36:37.729293+00:00 app[web.1]: return self.load_wsgiapp()
2019-11-13T00:36:37.729294+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 11, in <module>
2019-11-13T00:36:37.729296+00:00 app[web.1]: return util.import_app(self.app_uri)
2019-11-13T00:36:37.729298+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/site-packages/gunicorn/util.py", line 331, in import_app
2019-11-13T00:36:37.729300+00:00 app[web.1]: mod = importlib.import_module(module)
2019-11-13T00:36:37.729302+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.7/importlib/__init__.py", line 127, in import_module
2019-11-13T00:36:37.729305+00:00 app[web.1]: return _bootstrap._gcd_import(name[level:], package, level)
2019-11-13T00:36:37.729307+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:36:37.729309+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:36:37.729312+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 953, in _find_and_load_unlocked
2019-11-13T00:36:37.729313+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
2019-11-13T00:36:37.729315+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
2019-11-13T00:36:37.729317+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 983, in _find_and_load
2019-11-13T00:36:37.729319+00:00 app[web.1]: File "<frozen importlib._bootstrap>", line 965, in _find_and_load_unlocked
2019-11-13T00:36:37.729321+00:00 app[web.1]: ModuleNotFoundError: No module named 'django_form'
2019-11-13T00:36:37.729444+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [11] [INFO] Worker exiting (pid: 11)
2019-11-13T00:36:37.769198+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [4] [INFO] Shutting down: Master
2019-11-13T00:36:37.769467+00:00 app[web.1]: [2019-11-13 00:36:37 +0000] [4] [INFO] Reason: Worker failed to boot.
2019-11-13T00:36:37.845429+00:00 heroku[web.1]: Process exited with status 3

```

```

2019-11-13T00:36:29.256856+00:00 heroku[web.1]: State changed from crashed to starting
2019-11-13T00:36:35.313584+00:00 heroku[web.1]: Starting process with command `gunicorn django_form.wsgi --log-file -`
...
2019-11-13T00:36:37.729321+00:00 app[web.1]: ModuleNotFoundError: No module named 'django_form'

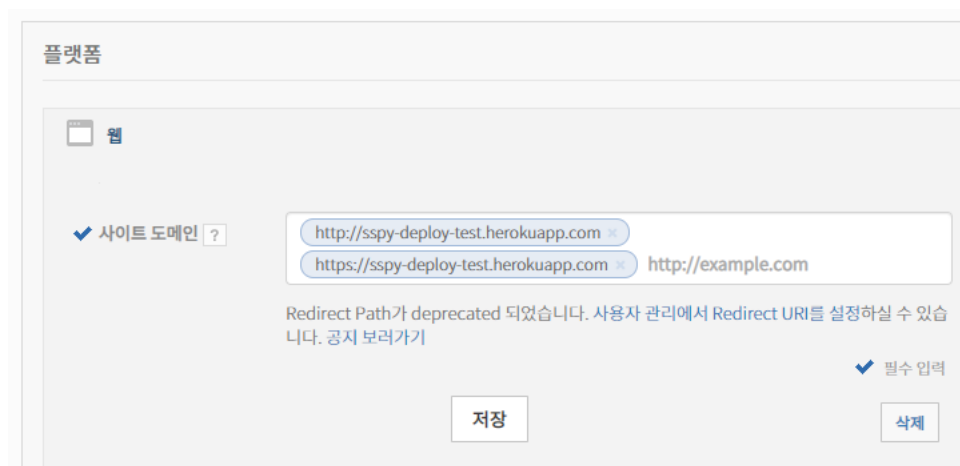
```

3.4. Social Login (Optional)

- 배포 이후에 소셜 로그인을 하게 되면 DoesNotExists 에러가 발생한다.
- admin 페이지에서 소셜 어플리케이션 등록을 새로 해주고 카카오 개발자 센터에도 Heroku 주소를 등록해주어야 한다.

kakao 개발자 센터

- [카카오 개발자 센터](#) - [설정](#) - [일반](#) - [플랫폼](#) - [웹](#)
 - [사이트 도메인](#) 에 자신의 heroku 주소 넣고 저장하기([http](#) , [https](#) 모두 넣어야 함)



- [사용자 관리](#) → [로그인 Redirect URI](#)
 - [/accounts/kakao/login/callback/](#) 추가

로그인 Redirect URI

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. REST API로 개발하는 경우 필수로 설정하여야 합니다.
- 최대 10개까지 설정할 수 있습니다.
- 추가 URI 등록은 [데브톡](#)으로 문의주세요.

https://sspy-deploy-test.herokuapp.com/accounts/kakao/login/callback/ ✕

http://sspy-deploy-test.herokuapp.com/accounts/kakao/login/callback/ ✕ http://example.com/oauth

✔ 저장되었습니다.

admin 페이지

- 소셜 계정 → 소셜 어플리케이션
 - 클라이언트 아이디: 일반 → REST API 키
 - 비밀 키: 일반 → 고급 → 코드

소셜 어플리케이션 변경

히스토리

제공자:

Kakao

이름:

justin

클라이언트 아이디:

85b3357b14654e9a4e8f44f046015d26

앱 아이디 또는 컨슈머 아이디

비밀 키:

81a44740e94444444444444444444444

API 비밀 키, 클라이언트 비밀 키, 또는 컨슈머 비밀 키

키:

키

Sites:

이용 가능한 sites



필터

모두 선택

선택된 sites

example.com

모두 제거

하나 이상을 선택하려면 "Control" 키, Mac은 "Command"키를 누르세요.

삭제

저장 및 다른 이름으로 추가

저장 및 편집 계속

저장

04. References

<https://developer.mozilla.org/ko/docs/Learn/Server-side/Django/Deployment>

[Configuring Django Apps for Heroku | Heroku Dev Center](https://devcenter.heroku.com/articles/django-app-configuration)

<https://devcenter.heroku.com/articles/django-app-configuration>

[Deploying Python Applications with Gunicorn | Heroku Dev Center](https://devcenter.heroku.com/articles/python-gunicorn)

<https://devcenter.heroku.com/articles/python-gunicorn>

[Heroku에 django 프로젝트 배포](https://dgkim5360.tistory.com/entry/deploy-django-project-on-heroku)

<https://dgkim5360.tistory.com/entry/deploy-django-project-on-heroku>