

13장. 배포(서비스)



파이썬 애니웨어 서비스

PyWeb 로그인 질문/답변 설문조사

Welcome~ Have a good Day!



오후 10:15:45

배포 과정(프로세스)

➤ 배포 프로세스(Process)

1. 프로젝트를 깃(Git)에 업로드 전 사전 작업
 - .gitignore 파일 , config/settings.py 변경
2. 깃(Git) 저장소에 업로드
3. 파이썬 애니웨어 Bash Console 설정
 - git clone으로 프로젝트 다운로드
 - venvs 디렉터리에 virtualenv 모듈 설치
 - 가상화 진입 후 프로젝트에 장고(django) 설치
 - migrate 명령 수행
 - createsuperuser 명령 수행
4. 파이썬 애니웨어 Web 설정
 - 새로운 앱 생성 > manual configuration
 - wsgi 설정 > config/wsgi.py에서 복사해 올
 - reload > 도메인 주소(본인ID.pythonanywhere.com)

깃(Git)에 올리기 전 사전 작업

➤ 깃(git)에 올리기 전 설정 변경

1. .gitignore 파일 만들기 – projects/pyweb/.gitignore
-> 관리하지 않아도 되는 파일 등록

```
.idea  
db.sqlite3  
*.pyc  
__pycache__
```

```
Untracked files:  
(use "git add <file>..."  
  .idea/  
  common/  
  config/  
  db.sqlite3  
  manage.py  
  pybo/  
  static/  
  templates/
```



```
Untracked files:  
(use "git add <file>..."  
  .gitignore  
  common/  
  config/  
  manage.py  
  pybo/  
  static/  
  templates/
```

깃(Git)에 올리기

➤ 깃(git)에 올리기 전 설정 변경

2. config/settings.py

```
DEBUG = False
```

```
ALLOWED_HOSTS = ['*']
```

DEBUG = True 일때 host 주소는 필요 없음

DEBUG = False 일때 host 주소 필수 설정해야 함

static 파일은 웹 서버(파이썬 애니웨어, NGIX 등)에서 설정 해 주어야 함
(정적 파일은 매번 랜더링 할 필요가 없어서 웹서버에서 처리하고, 템플릿은 요청시마다 매번 랜더링 되므로 동적파일이라 할 수 있다.)

배포 후 django에서는 개발에서는 DEBUG = True 로 **변경**해야 static 업로드 가능함.

깃(Git)에 올리기

➤ 깃(git)에 올리기(로컬 : 명령프롬프트)

원격 Repository(저장소) 경로 : <http://github.com/본인ID/pyweb.git>

```
C:\projects\pyweb> git init
```

```
C:\projects\pyweb> git add *
```

```
C:\projects\pyweb> git commit -m "pyweb 전체파일"
```

```
C:\projects\pyweb> git remote add origin 원격 저장소 경로
```

```
C:\projects\pyweb> git push -u origin master
```

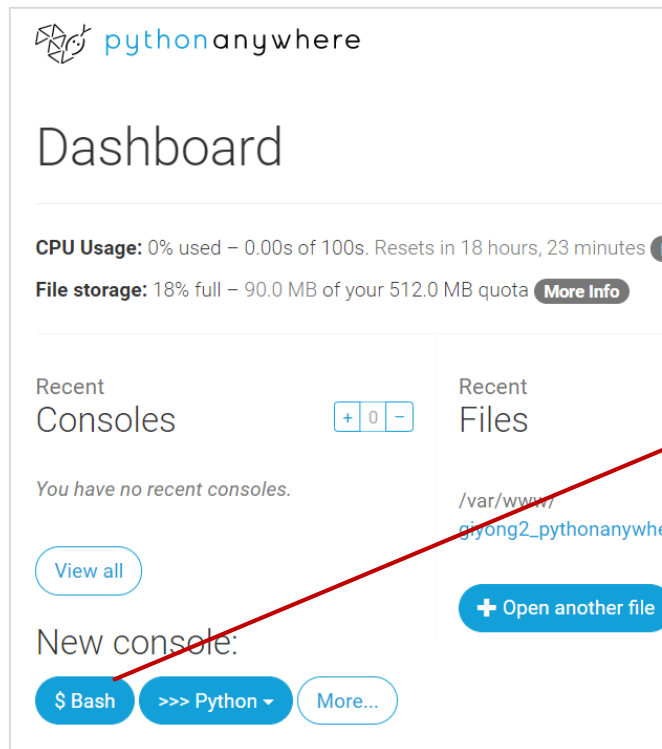
파이썬 애니웨어 서비스

- 파이썬 애니웨어(python anywhere) 서비스

파이썬 언어를 기반으로 하는 웹 호스팅 서비스이다.

회원 가입

Sign up > Creat a Beginner account > log in(이메일, 비밀번호)



Bash 콘솔은 리눅스 운영체제임



파이썬 애니웨어 서비스

● 리눅스 명령어 익히기

명령어	설 명
ls	목록 보기
pwd	현재 위치(경로)
cd ~/	루트 디렉터리로 이동
mkdir	디렉터리 만들기
rm -rf 디렉터리	디렉터리 삭제(내부 모두 삭제)
clear	화면 지우기
. 파일이름	파일 실행

파이썬 애니웨어 Bash - console

➤ 파이썬 애니웨어 Bash 콘솔 사용 - 리눅스 운영체제

~ \$ ls -> **목록 보기**

~ \$ pwd -> **현재 위치(경로)**

~ \$ git clone http://github.com/kiyongee2/pyweb -> **깃에서 프로젝트 다운로드**

~ \$ cd pyweb -> **pyweb으로 이동**

~/pyweb (master)\$ **virtualenv venvs --python=python3.8**

~/pyweb (master)\$ cd venvs

~/pyweb/venvs (master)\$ cd bin

venvs 디렉터리에 virtualenv 모듈
설치하기

파이썬 애니웨어 Bash - console

➤ 파이썬 애니웨어 Bash 콘솔 사용 - 리눅스 운영체제

~/pyweb/venvs/bin (master)\$. activate -> 가상으로 진입(점 한칸 띄고)

(venvs) ~/pyweb/venvs/bin (master)\$ cd ~/pyweb -> pyweb으로 이동

(venvs) ~/pyweb (master)\$ **pip install django** -> 장고 설치

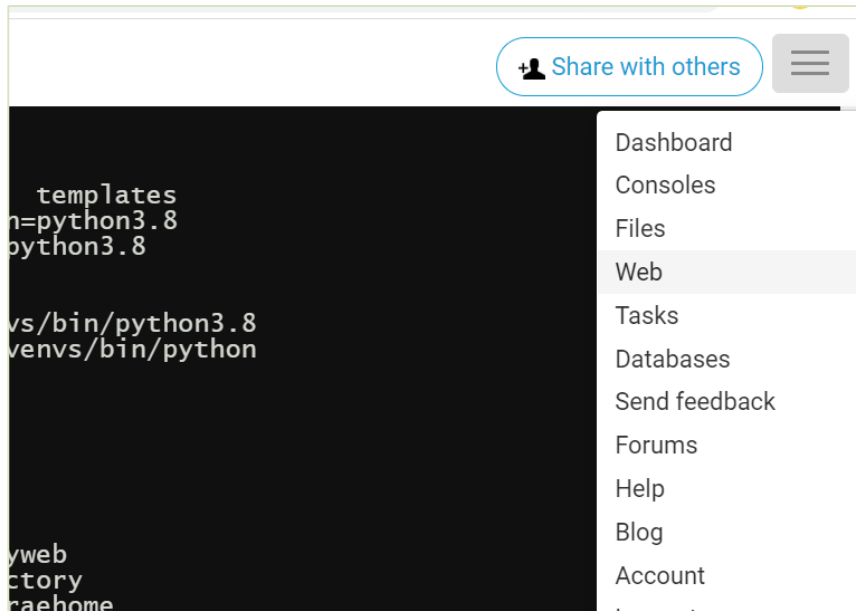
(venvs) ~/pyweb (master)\$ python manage.py migrate -> DB 설정

(venvs) ~/pyweb (master)\$ python manage.py createsuperuser -> admin 설정

- ▶ 파일 실행 방법 : 리눅스는 윈도우와 다르게 점(.) 한칸 띄고 파일이름을 작성
- ▶ 절대 경로 이동 : ~/디렉터리 이름

파이썬 애니웨어 Web 설정

➤ Bash 세팅 후 > Web으로 이동



Add a new wep app > manual configuration > python3.8

파이썬 애니웨어 Web 설정

➤ Web 설정

Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)

Working directory: [/home/gyiong2/](#)

WSGI configuration file: [/var/www/gyiong2_pythonanywhere_com_wsgi.py](#)

Python version: 3.8 

①

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.


②

[/home/gyiong2/graehome/venvs](#)

 Start a console in this virtual

1번 설정후 반드시 입력해야 함

③

 Reload gyiong2.pythonanywhere.com

파이썬 애니웨어 Web 설정

➤ Wsgi(Web Server GateWay Interface) 설정

웹서버와 웹 애플리케이션의 인터페이스를 위한 파이썬 프레임워크다.

검정색 코드는 config/wsgi.py에서 복사해옴

빨강색 코드는 Static 제어 관련 설정임

```
import os
import sys
```

```
path = "/home/giyong2/pyweb"
if path not in sys.path:
    sys.path.append(path)
```

```
from django.contrib.staticfiles.handlers import StaticFilesHandler
from django.core.wsgi import get_wsgi_application
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'config.settings')
```

```
application = StaticFilesHandler(get_wsgi_application())
```

```
import os
```

```
from django.core.wsgi import get_wsgi_application
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'config.settings')
```

```
application = get_wsgi_application()
```

서버에 업데이트 반영(재 업로드)

1. Comment 모델을 admin에 등록하기

```
from django.contrib import admin
from .models import Question, Answer, Comment

admin.site.register(Question)
admin.site.register(Answer)
admin.site.register(Comment)
```

2. 변경된 admin.py를 깃에 올리기

서버에 업데이트 반영하기

3. Bash 콘솔로 가져오기

(venvs) 09:28 ~/pyweb (master)\$ git pull -> 가져오기(병합하기)


```
06:52 ~ $ ls
README.txt  pyweb
06:52 ~ $ cd pyweb
06:52 ~/pyweb (master)$ git pull
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 14 (delta 8), reused 14 (delta 8), pack-reused 0
Unpacking objects: 100% (14/14), done.
From http://github.com/kiyongee2/pyweb
   e02bf06..4de642c  master    -> origin/master
Updating e02bf06..4de642c
Fast-forward
 pybo/urls.py          | 2  ++
 pybo/views/base_views.py | 3  +++
 static/pybo/css/popup.css | 24 +++++
 templates/pybo/index.html | 4  +++-
 templates/pybo/notice.html | 18 +++++
5 files changed, 50 insertions(+), 1 deletion(-)
create mode 100644 static/pybo/css/popup.css
create mode 100644 templates/pybo/notice.html
```

서버에 업데이트 반영하기

4. Bash Console에서 작업후 Web에서 Reload 실행

Configuration for giyong2.pythonanywhere.com

Reload:

 Reload giyong2.pythonanywhere.com

Django 관리자 페이지

Django 관리

사이트 관리

PYBO

Answers

+ 추가 ✎ 변경

Comments

+ 추가 ✎ 변경

Questions

+ 추가 ✎ 변경

인증 및 권한

그룹

+ 추가 ✎ 변경

사용자(들)

+ 추가 ✎ 변경

서버에 업데이트 반영하기

5. Bash Console 삭제

Delete:

Deleting this web app will remove your web app from the Internet, but will not remove your code. The WSGI configuration file will be backed up.

 Delete giyong2.pythonanywhere.com