

nginx + uWSGI + dJango + python 프로젝트 구축

X 미완 X

설치환경 ::

OS X El Capitan
버전 10.11.3(15D21)

패러렐즈11 +



Python3.5

장고 버전정보 확인

```
sodarain730@ubuntu:~$ python3 -c "import django; print(django.get_version())"  
1.9.1  
sodarain730@ubuntu:~$
```

참고 : <https://www.youtube.com/watch?v=ZIGBMst-8ls>

1. 설치 할 계정 생성

: sudo adduser <계정이름>
비밀번호 입력

```
sodara730@ubuntu:~$ sudo adduser sodara730
sodara730@ubuntu:~$ pwd
Enter new UNIX password:
Retype new UNIX password:
```

2. 계정 변경

: su - <계정이름>
비밀번호 입력

3. 유저를 추가하고 나면 이 유저가 슈퍼권한을 사용할 수 있도록 해야한다.

: su - <추가 한 계정명과 원래계정>

: sudo visudo

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
sodara730 ALL=(ALL:ALL) ALL
```

<-<계정명> \t ALL=(ALL:ALL) 쓰고 ctrl+X , y 로 저장하고 빠져나오기

4. apt source list를 보면...(sudo vi /etc/apt/sources.list)

```
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ trusty universe
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty universe
deb http://us.archive.ubuntu.com/ubuntu/ trusty-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty-updates universe
```

주소들이 us. 으로 시작되는 것을 확인 할 수 있다. 리눅스 패키지를 설치 할 때 미국서버에서 다운 받는다는 뜻. 안해도 되긴 하지만... 미국 서버에서 다운받는 것은 느리므로(10kb) 우리나라에서 서비스 하는 곳(100~200kb)으로 변경 : q! 로 빠져나오기

```
sodara730@ubuntu:~$ sudo sed -i 's/us.archive.ubuntu.com/ftp.neowiz.com/g' /etc/apt/sources.list
sodara730@ubuntu:~$ sudo sed -i 's/security.archive.ubuntu.com/ftp.neowiz.com/g' /etc/apt/sources.list
```

sed라는 치환명령어를 통해서 us.archive.ubuntu.com 을 ftp.neowiz.com으로 변경

바뀐 주소 확인

```
sudo vi /etc/apt/sources.list
```

```
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://ftp.neowiz.com/ubuntu/ trusty universe
deb-src http://ftp.neowiz.com/ubuntu/ trusty universe
deb http://ftp.neowiz.com/ubuntu/ trusty-updates universe
deb-src http://ftp.neowiz.com/ubuntu/ trusty-updates universe
```

5. sudo apt-get update 테스트

```
sodara730@ubuntu:~$ sudo apt-get update
```

```
Fetches 32.3 MB in 13s (2,356 kB/s)
W: Failed to fetch http://ftp.neowiz.com/ubuntu/dists/trusty-updates/main/source/Sources Hash Sum mismatch

W: Failed to fetch http://ftp.neowiz.com/ubuntu/dists/trusty-updates/main/binary-amd64/Packages Hash Sum mismatch

W: Failed to fetch http://ftp.neowiz.com/ubuntu/dists/trusty-updates/universe/binary-amd64/Packages Hash Sum mismatch

W: Failed to fetch http://ftp.neowiz.com/ubuntu/dists/trusty-updates/main/binary-i386/Packages Hash Sum mismatch

W: Failed to fetch http://ftp.neowiz.com/ubuntu/dists/trusty-updates/universe/binary-i386/Packages Hash Sum mismatch

E: Some index files failed to download. They have been ignored, or old ones used instead.
```

확인하니 뭔가.. Failed가 많네요;
전에 했던 것 같아서 일단 넘어갔습니다...

6. 패키지 설치 `sudo apt-get install vim git build-essential python python-dev python-pip nginx uwsgi uwsgi-plugin-python`

```
18 sudo apt-get install vim git build-essential python3 python3-dev python3-pip nginx uwsgi uwsgi-plugin-python
```

vim

git(github에서 소스를 가져올 때 사용하는 패키지)

build-essential(빌드를 하는데 꼭 필요한 패키지(gcc등)를 패키징 해서 배포하는 패키지)

python3

python-dev 파이썬 개발키트

python3-pip 파이썬 패키지 설치 관리 패키지

nginx

uwsgi

uwsgi-plugin-python uwsgi를 파이썬과 연동하여 사용할 수 있는 플러그인(서로 인터페이스를 맞춰 준다고 생각하면 된다)

-기타: vim 설정 `sodara@indjango@ubuntu:~$ vim .vimrc`

```
set nu
syntax on
~
~
~
~
~
:wq
```

적용이 안되네요 .. 나중에 아래 링크보고 다시 하기로...

<https://medium.com/sunhyoups-story/vim-에디터-이쁘게-사용하기-5b6b8d546017#2moh26vng>

7. 전에 설치했던 virtualenv(sudo pip install virtualenv)를 사용하여 환경변수를 위한 가상 환경을 만들기

```
sodaraindjango@ubuntu:~$ virtualenv django_env
New python executable in /home/sodaraindjango/django_env/bin/python
Installing setuptools, pip, wheel...done.
sodaraindjango@ubuntu:~$ ls
django_env  examples.desktop
sodaraindjango@ubuntu:~$ pwd
/home/sodaraindjango
```

장고만을 위한 계정을 따로 만들었으므로 그냥 홈 디렉토리에 만들었습니다.

8. 가상환경 활성화 명령어: source <env이름>/bin/activate

```
sodaraindjango@ubuntu:~$ source django_env/bin/activate
(django_env) sodaraindjango@ubuntu:~$ deactivate
sodaraindjango@ubuntu:~$ source django_env/bin/activate
(django_env) sodaraindjango@ubuntu:~$
```

<-참고로 비 활성화는 deactivate

—— 이제부터 가급적 sudo는 사용하지 말기 ——

sudo pip을 사용하면 pip가 본래의 것으로 사용되므로 주의

```
(flask_env)flaskadmin@ubuntu:~/flask_blog$ sudo which pip
/usr/bin/pip
(flask_env)flaskadmin@ubuntu:~/flask_blog$ which pip
/home/flaskadmin/flask_env/bin/pip
```

여기서 부터… 다른 자료를 참고하였습니다.

<http://blog.yong-hee.com/97>

9. 이전에 만든 가상환경안에 myFirstWebApp 프로젝트 생성

```
(django_env) sodaraindjango@ubuntu:~$ django-admin.py startproject myFirstWebApp
(django_env) sodaraindjango@ubuntu:~$ ls
django_env  examples.desktop  myFirstWebApp
(django_env) sodaraindjango@ubuntu:~$ cd myFirstWebApp/
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp$ ls
manage.py  myFirstWebApp
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp$ cd myFirstWebApp/
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp/myFirstWebApp$ ls
__init__.py  settings.py  urls.py  wsgi.py
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp/myFirstWebApp$
```

```
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp$ python3 manage.py startapp hello
ld
(django_env) sodaraindjango@ubuntu:~/myFirstWebApp$ ls
helloworld  manage.py  myFirstWebApp
```