가상 환경부터 시작합니다

uWSGI is a WSGI implementation. In this tutorial we will set up uWSGI so that it creates a Uni
socket, and serves responses to the web server via the WSGI protocol. At the end, our complet
of components will look like this:

```
the web client <-> the web server <-> the socket <-> uwsgi <-> Django
```

Before you start setting up uWSGI

virtualenv

Make sure you are in a virtualenv for the software we need to install (we will describe how to in
system-wide uwsgi later):

```
virtualenv uwsgi-tutorial
cd uwsgi-tutorial
source bin/activate
```

Django

Install Django into your virtualenv, create a new project, and `cd` into the project:

```
pip install Django
django-admin.py startproject mysite
cd mysite
```

About the domain and port

In this tutorial we will call your domain `example.com`. Substitute your own FQDN or IP address.

Throughout, we'll be using port 8000 for the web server to publish on, just like the Django runserver does by default. You can use whatever port you want of course, but I have chosen this one so it doesn't conflict with anything a web server might be doing already.

Basic uWSGI installation and configuration

```
goohoon@goohoon-VirtualBox:~$ ls
Env               fc      mysite2  공개      문서      비디오  음악
examples.desktop  mysite  pip      다운로드  바탕화면  사진    템플릿
goohooh@goohooh-VirtualBox:~$ virtualenv uwsgi-tutorial
New python executable in /home/goohooh/uwsgi-tutorial/bin/python
Installing setuptools, pip, wheel...done.
goohooh@goohooh-VirtualBox:~$ cd uwsgi-tutorial
goohooh@goohooh-VirtualBox:~/uwsgi-tutorial$ source bin/activate
```

가상환경 실행

application, feed it requests from web clients (such as browsers) and return responses.

A Web Server Gateway Interface - WSGI - does this job. WSGI is a Python standard.

uWSGI is a WSGI implementation. In this tutorial we will set up uWSGI so that it creates a Unix socket, and serves responses to the web server via the WSGI protocol. At the end, our complete stack of components will look like this:

```
the web client <-> the web server <-> the socket <-> uwsgi <-> Django
```

Before you start setting up uWSGI

virtualenv

Make sure you are in a virtualenv for the software we need to install (we will describe how to install a system-wide uwsgi later):

```
virtualenv uwsgi-tutorial
cd uwsgi-tutorial
source bin/activate
```

튜토리얼 생성

Django

Install Django into your virtualenv, create a new project, and `cd` into the project:

```
pip install Django
django-admin.py startproject mysite
cd mysite
```

간단한 헬로월드 찍기

About the domain and port

In this tutorial we will call your domain `example.com`. Substitute your own FQDN or IP address.

Throughout, we'll be using port 8000 for the web server to publish on, just like the Django runserver does by default. You can use whatever port you want of course, but I have chosen this one so it doesn't conflict with anything a web server might be doing already.

Basic uWSGI installation and configuration

```
goohooh@goohooh-VirtualBox: ~/uwsgi-tutorial/mysite
GNU coreutils 홈 페이지: <http://www.gnu.org/software/coreutils/>
GNU 소프트웨어 사용에 관련된 전반적인 도움을 얻기: <http://www.gnu.org/
Report rm translation bugs to <http://translationproject.org/team/>
For complete documentation, run: info coreutils 'rm invocation'
goohooh@goohooh-VirtualBox:~$ rm -r uwsgi-tutorial/
goohooh@goohooh-VirtualBox:~$ ls
Env              fc      mysite2  공개      문서      비디오  음악
examples.desktop  mysite  pip      다운로드  바탕화면  사진    템플릿
goohooh@goohooh-VirtualBox:~$ virtualenv uwsgi-tutorial
New python executable in /home/goohooh/uwsgi-tutorial/bin/python
Installing setuptools, pip, wheel...done.
goohooh@goohooh-VirtualBox:~$ cd uwsgi-tutorial/
goohooh@goohooh-VirtualBox:~/uwsgi-tutorial$ source bin/activate
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial$ pip3 inst
o
Requirement already satisfied (use --upgrade to upgrade): Django in /us
ib/python3.4/dist-packages
Cleaning up...
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial$ django-ad
artproject mysite
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial$ cd mysite
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial/mysite$ ls
manage.py  mysite
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial/mysite$
```

Install Django into your virtualenv, create a new project, and `cd` into the project:

```
pip install Django
django-admin.py startproject mysite
cd mysite
```

About the domain and port

In this tutorial we will call your domain `example.com`. Substitute your own FQDN or IP address.

Throughout, we'll be using port 8000 for the web server to publish on, just like the Django runserver does by default. You can use whatever port you want of course, but I have chosen this one so it doesn't conflict with anything a web server might be doing already.

Basic uWSGI installation and configuration

Install uWSGI into your virtualenv

```
pip install uwsgi
```

uwsgi 설치

를 해야하는데 에러

Of course there are other ways to install uWSGI, but this one is as good as any. Remember that you will need to have Python development packages installed. In the case of Debian, or Debian-derived systems such as Ubuntu, what you need to have installed is `pythonX.Y-dev`, where X.Y is your version of Python.

Basic test

Create a file called `test.py`:

```
# test.py
def application(env, start_response):
    start_response('200 OK', [('Content-Type','text/html')])
    return [b"Hello World"] # python3
    #return ["Hello World"] # python2
```

Note

Take into account that Python 3 requires `bytes()`.

```
goohooh@goohooh-VirtualBox: ~/uwsgi-tutorial/mysite
dblog/mongodblog_plugin.o plugins/router_rewrite/router_rewrite.o plugin
_http/router_http.o plugins/logfile/logfile.o plugins/router_cache/route
o plugins/rawrouter/rawrouter.o plugins/router_static/router_static.o pl
lrouter/sslrouter.o plugins/spooler/spooler_plugin.o plugins/cheaper_bus
eaper_busyness.o plugins/symcall/symcall_plugin.o plugins/transformation
tofile.o plugins/transformation_gzip/gzip.o plugins/transformation_chunk
ed.o plugins/transformation_offload/offload.o plugins/router_memcached/r
mcached.o plugins/router_redis/router_redis.o plugins/router_hash/router
plugins/router_expires/expires.o plugins/router_metrics/plugin.o plugins
rmation_template/tt.o plugins/stats_pusher_socket/plugin.o -lpthread -lm
ic -ldl -lz -lssl -lcrypto -lexpat -lpthread -ldl -lutil -lm -lpython3.4
t

*** error linking uWSGI ***

----------------------------------------
Cleaning up...
Command /usr/bin/python3 -c "import setuptools, tokenize;__file__='/tmp/
d_goohooh/uwsgi/setup.py';exec(compile(getattr(tokenize, 'open', open)(
 read().replace('\r\n', '\n'), __file__, 'exec'))" install --record /tmp
4xytv-record/install-record.txt --single-version-externally-managed --co
iled with error code 1 in /tmp/pip_build_goohooh/uwsgi
Storing debug log for failure in /home/goohooh/.pip/pip.log
(uwsgi-tutorial) goohooh@goohooh                    /mysite$
```

로그..로그를 보자!

터미널

pip.log (~/.pip) - VIM

```
436
437 [x86_64-linux-gnu-gcc -pthread] plugins/http/keepalive.o
438
439 [x86_64-linux-gnu-gcc -pthread] plugins/http/https.o
440
441 [x86_64-linux-gnu-gcc -pthread] plugins/http/spdy3.o
442
443 [x86_64-linux-gnu-gcc -pthread] plugins/ugreen/ugreen.o
444
445 [x86_64-linux-gnu-gcc -pthread] plugins/signal/signal_plugin.o
446
447 [x86_64-linux-gnu-gcc -pthread] plugins/syslog/syslog_plugin.o
448
449 [x86_64-linux-gnu-gcc -pthread] plugins/rsyslog/rsyslog_plugin.o
450
451 [x86_64-linux-gnu-gcc -pthread] plugins/logsocket/logsocket_plugin.o
452
453 [x86_64-linux-gnu-gcc -pthread] plugins/router_uwsgi/router_uwsgi.o
454
455 [x86_64-linux-gnu-gcc -pthread] plugins/router_redirect/route
    r/bin/ld: cannot open output file /usr/bin/uwsgi: 허가 거부
456
457 collect2: error: ld returned 1 exit status
458
459
460
461 [x86_64-linux-gnu-gcc -pthread] plugins/router_basicauth/router_basicauth.o
462
463 [x86_64-linux-gnu-gcc -pthread] plugins/zergpool/zergpool.o
464
465 [x86_64-linux-gnu-gcc -pthread] plugins/redislog/redislog_plugin.o
466
467 [x86_64-linux-gnu-gcc -pthread] plugins/mongodblog/mongodblog_plugin.o
468
469 [x86_64-linux-gnu-gcc -pthread] plugins/router_rewrite/router_rewrite.o
470
471 [x86_64-linux-gnu-gcc -pthread] plugins/router_http/router_http.o
472
473 [x86_64-linux-gnu-gcc -pthread] plugins/logfile/logfile.o
474
475 [x86_64-linux-gnu-gcc -pthread] plugins/router_cache/router_cache.o
476
477 [x86_64-linux-gnu-gcc -pthread] plugins/rawrouter/rawrouter.o
478
479 [x86_64-linux-gnu-gcc -pthread] plugins/router_static/router_static.o
480
481 [x86_64-linux-gnu-gcc -pthread] plugins/sslrouter/sslrouter.o
482
483 [x86_64-linux-gnu-gcc -pthread] plugins/spooler/spooler_plugin.o
484
485 [x86_64-linux-gnu-gcc -pthread] plugins/cheaper_busyness/cheaper_busyness.o
486
487 [x86_64-linux-gnu-gcc -pthread] plugins/symcall/symcall_plugin.o
```

허가?권한 문제인가?

Search

t, and `cd` into the project:

ubstitute your own FQDN or IP address.

r to publish on, just like the Django runserver

ve chosen this one so it doesn't

nfiguration

is one is as good as any. Remember that you

led. In the case of Debian, or Debian-derived

ed is `pythonX.Y-dev`, where X.Y is your version

Dubeolsik 한자 전용

/html'))

Take into account that Python 3 requires `bytes()`.

goohooh@goohooh-VirtualBox:

```
dblog/mongodblog_plugin.o plugins/
_http/router_http.o plugins/logfil
o plugins/rawrouter/rawrouter.o pl
lrouter/sslrouter.o plugins/spoole
eaper_busyness.o plugins/symcall/s
tofile.o plugins/transformation_gz
ed.o plugins/transformation_offloa
mcached.o plugins/router_redis/rou
plugins/router_expires/expires.o p
rmation_template/tt.o plugins/stat
ic -ldl -lz -lssl -lcrypto -lexpat
t

*** error linking uWSGI ***

----------------------------------
Cleaning up...
Command /usr/bin/python3 -c "impor
d_goohooh/uwsgi/setup.py';exec(com
.read().replace('\r\n', '\n'), __f
4xyiv-record/install-record.txt --
lled with error code 1 in /tmp/pip
Storing debug log for failure in /
(uwsgi-tutorial) goohooh@goohooh-V
```

(100%)

Search

goohooh@goohooh-VirtualBox: ~/mysite

```
*** Python threads support is disabled. You can enable it with --enable-thre
***
Python main interpreter initialized at 0x1f3cea0
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 72768 bytes (71 KB) for 1 cores
*** Operational MODE: single process ***
WSGI app 0 (mountpoint='') ready in 0 seconds on interpreter 0x1f3cea0 pid:
(default app)
*** uWSGI is running in mu
spawned uWSGI worker 1 (an                1)
pid: 9474|app: 0|req: 1/1                 bytes} [Wed Jan 27
:48 2016] GET / => genera                 1.1 200) 1 headers i
bytes (1 switches on core
^C(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/uwsgi-tutorial/mysite$ cd ..
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~$ ls
env                 mysite   uwsgi-tutorial  문서      사진
examples.desktop    mysite2  공개            바탕화면  음악
mc                  pip      다운로드        비디오    템플릿
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~$ cd mysite
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ ls
db.sqlite3  helloworld  manage.py  mysite  simblog
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ uwsgi --http :8000 --m
e mysite.wsgi
```

sudo로 설치하니 잘됨

zilla Firefox

pyenv를 이용하여 여... × | * student | fastcamp... × | Setting up Django a... ×

localhost:8000

Hello World

터미널

잘 실행됨

pyenv를 이용하여 여... | * student | fastcamp... | Setting up Django a... | Goohooh's Blog

localhost:8000

Welcome!

post

- Baseball
- Book
- Music

[Baseball] 뱀직구 인터뷰

창용니뮤ㅠㅠ...

[Book] TCP/IP

너란녀석 다 읽어주겠어

[Music] 위아래위위아래

오오오오오~

[Baseball] 야구하고 싶다

피칭피칭피칭

[Book] 소설

읽은지 오래됐다

footer....

명령문만 바꾸니
기존 사이트도 실행됨

Dubeolsik 한자 진용

```
goohooh@goohooh-VirtualBox: ~/mysit
your processes number limit is 7921
your memory page size is 4096 bytes
detected max file descriptor number: 10
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable
uWSGI http bound on :8000 fd 4
spawned uWSGI http 1 (pid: 9519)
uwsgi socket 0 bound to TCP address 127
Python version: 3.4.3 (default, Oct 14
*** Python threads support is disabled.
***
Python main interpreter initialized at
your server socket listen backlog is li
your mercy for graceful operations on w
mapped 72768 bytes (71 KB) for 1 cores
*** Operational MODE: single process **
WSGI app 0 (mountpoint='') ready in 1 s
(default app)
*** uWSGI is running in multiple interp
spawned uWSGI worker 1 (and the only) (
[pid: 9518|app: 0|req: 1/1] 127.0.0.1 (
8:12 2016] GET / => generated 1156 byte
n 88 bytes (1 switches on core 0)
```

Point your browser at the server; if the site appears, it means uWSGI is able to serve your Django application from your virtualenv, and this stack operates correctly:

```
the web client <-> uWSGI <-> Django
```

Now normally we won't have the browser speaking directly to uWSGI. That's a job for the webserver, which will act as a go-between.

Basic nginx

Install nginx

```
sudo apt-get install nginx
sudo /etc/init.d/nginx start    # start nginx
```

이제 nginx 설치

And now check that the nginx is serving by visiting it in a web browser on port 80 - you should get a message from nginx: "Welcome to nginx!". That means these components of the full stack are working together:

```
the web client <-> the web server
```

If something else *is* already serving on port 80 and you want to use nginx there, you'll have to reconfigure nginx to serve on a different port. For this tutor[illegible]ng port 8000.

Dubeolsik 한자 진용

Configure nginx for your site

You will need the `uwsgi_params` file, which is available in the `nginx` directory of the uWSGI distribution, or from https://github.com/nginx/nginx/blob/master/conf/uwsgi_params

Copy it into your project directory. In a moment we will tell nginx to refer to it.

Now create a file called mysite_nginx.conf, and put this in it:

```
# mysite_nginx.conf

# the upstream component nginx needs to connect to
```

```
goohooh@goohooh-VirtualBox: ~/mysite
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 72768 bytes (71 KB) for 1 cores
*** Operational MODE: single process ***
WSGI app 0 (mountpoint='') ready in 1 seconds on interpreter 0x7f1e90 pid: 9518
(default app)
*** uWSGI is running in multiple interpreter mode ***
spawned uWSGI worker 1 (and the only) (pid: 9518, cores: 1)
[pid: 9518|app: 0|req: 1/1] 127.0.0.1 () {38 vars in 661 bytes} [Wed Jan 27 10:5
8:12 2016] GET / => generated 1156 bytes in 123 msecs (HTTP/1.1 200) 2 headers i
n 88 bytes (1 switches on core 0)
^C(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ sudo apte-get install ng
x^C
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ ls
db.sqlite3  helloworld  manage.py  mysite  simblog
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ sudo apt-get install nginx
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
nginx 패키지는 이미 최신 버전입니다.
0개 업그레이드, 0개 새로 설치, 0개 제거 및 245개 업그레이드 안 함.
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ sudo /etc/init.d/nginx sta
rt
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$
```

ox

... × # * student | fastcamp... × Setting up Django a... × Welcome to nginx! ×

Search

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working.
Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

원할한 실행

Dubeolsik 한자 전용

web server with uWSGI and nginx — uWSGI 2.0 documentation - Mozilla Firefox

inx/uwsgi_param... × | Problem loading page ×

/latest/tutorials/Django_and_ng | Search

the webserver, which will act as a go-between.

Basic nginx

Install nginx

```
sudo apt-get install nginx
sudo /etc/init.d/nginx start    # start nginx
```

And now check that the nginx is serving by visiting it in a web br
you should get a message from nginx: "Welcome to nginx!". That
components of the full stack are working together:

```
the web client <-> the web server
```

If something else *is* already serving on port 80 and you want to u
have to reconfigure nginx to serve on a different port. For this tutorial though, we'
going to be using port 8000.

Configure nginx for your site

You will need the `uwsgi_params` file, which is available in the `nginx` directory of th
uWSGI distribution, or from https://github.com/nginx/nginx/blob/master
/conf/uwsgi_params

Copy it into your
to refer to it.

Now create a file

```
# mysite_nginx.conf

# the upstream component nginx needs to connect to
upstream django {
    # server unix:///path/to/your/mysite/mysite.sock; # for a file socket
    server 127.0.0.1:8001; # for a web port socket (we'll use this first)
}

# configuration of the server
```

uwsgi

```
 3 uwsgi_par
 4 uwsgi_par
 5
 6 uwsgi_par
 7 uwsgi_par
 8 uwsgi_par
 9 uwsgi_par
10 uwsgi_par
11 uwsgi_par
12
13 uwsgi_par
14 uwsgi_par
15 uwsgi_par
16 uwsgi_par
~
~
~
~
~
~
~
"uwsgi_params
"uwsgi_params
계속하려면 엔
```

uwsgi_params 파일을 아래 링크 내용으로 복붙

uwsgi_params = (/etc/nginx) - VIM

```
 1 uwsgi_param  QUERY_STRING       $query_string;
 2 uwsgi_param  REQUEST_METHOD     $request_method;
 3 uwsgi_param  CONTENT_TYPE       $content_type;
 4 uwsgi_param  CONTENT_LENGTH     $content_length;
 5
 6 uwsgi_param  REQUEST_URI        $request_uri;
 7 uwsgi_param  PATH_INFO          $document_uri;
 8 uwsgi_param  DOCUMENT_ROOT      $document_root;
 9 uwsgi_param  SERVER_PROTOCOL    $server_protocol;
10 uwsgi_param  REQUEST_SCHEME     $scheme;
11 uwsgi_param  HTTPS              $https if_not_empty;
12
13 uwsgi_param  REMOTE_ADDR        $remote_addr;
14 uwsgi_param  REMOTE_PORT        $remote_port;
15 uwsgi_param  SERVER_PORT        $server_port;
16 uwsgi_param  SERVER_NAME        $server_name;
17
18 #uwsgi_param    QUERY_STRING       $query_string;
19 #uwsgi_param    REQUEST_METHOD     $request_method;
20 #uwsgi_param    CONTENT_TYPE       $content_type;
21 #uwsgi_param    CONTENT_LENGTH     $content_length;
22
23 #uwsgi_param    REQUEST_URI    $request_uri;
"uwsgi_params" [읽기 전용] 32L, 1209C
```

사알짝 다른 변경이 있음

web server with uWSGI and nginx — uWSGI 2.0 documentation - Mozilla Firefox

oblem loading page ×

/latest/tutorials/Django_and_ng

Search

You will need the `uwsgi_params` file, which is available in the `nginx` directory of the uWSGI distribution, or from https://github.com/nginx/nginx/blob/master/conf/uwsgi_params

to refer to it.

mysite_nginx.conf파일 생성 후
아래 내용 복붙

```
# mysite_nginx.conf

# the upstream component nginx needs to connect to
upstream django {
    # server unix:///path/to/your/mysite/mysite.sock; # for a file socket
    server 127.0.0.1:8001; # for a web port socket (we'll use this first)
}

# configuration of the server
server {
    # the port your site will be served on
    listen      8000;
    # the domain name it will serve for
    server_name .example.com; # substitute your machine's IP address or FQDN
    charset     utf-8;

    # max upload size
    client_max_body_size 75M;   # adjust to taste

    # Django media
    location /media  {
        alias /path/to/your/mysite/media;  # your Django project's media files -
    }

    location /static {
        alias /path/to/your/mysite/static; # your Django project's static files -
    }

    # Finally, send all non-media requests to the Django server.
    location / {
        uwsgi_pass  django;
        include     /path/to/your/mysite/uwsgi_params; # the uwsgi_params file yo
    }
}
```

This conf file tells nginx to serve up media and static files from the filesystem, as well as handle requests that require Django's intervention. For a large deployment it is

mysite_nginx.conf + (~/mysite) - VIM

```
1 upstream django {
2     server unix://~/mysite/mysite.sock
3     # server unix:///path/to/your/mysite/mysite.sock; # for a file soc
4     server 127.0.0.1:8001; # for a web port socket (we'll use this fir
5 }
6
7 # configuration of the server
8 server {
9     # the port your site will be served on
10    listen      8000;
11    # the domain name it will serve for
12    server_name .example.com; # substitute your machine's IP address o
13    charset     utf-8;
14
15    # max upload size
16    client_max_body_size 75M;   # adjust to taste
17
18    # Django media
19    location /media  {
20        alias ~/mysite/media;  # your Django project's media files - a
   s required
21    }
22
-- 끼워넣기 (붙이기) --
```

verflow - Mozilla Firefox

no internal routi... | Need help for bu... | rebuild uwsgi wit...

ild-uwsgi-with-pcre | Search

active | oldest | votes

talls the python egg. You need to rebuild the uwsgi binary

e3-dev

gi and then run the pip installer again.

d a bunch of compiler messages. The last message

n ##################

ation ##############

answered Mar 25 '14 at 20:48

user1046783

javascript d3.js

Linked

0 How to start uwsgi with flask ar

0 How do I prevent this bad gatev using nginx, uwsgi, and flask?

Related

2 PHP PCRE (regex) doesn't supp UTF-8?

5 uwsgi socket permissions

15 pip-installed uWSGI ./python_plu error

1 uwsgi: find: `/etc/uwsgi/s-enable such file or directory

0 How do turn routing support for u heroku?

1 How to install Git with PCRE sup OSX with homebrew?

1 debian uwsgi not built with ssl su

0 How to rebuild uwsgi with xml =

1 uWSGI with pcre support

1 uwsgi : pcre jit option in the conf disabled

Hot Network Questions

Please explain, "Asymmetric is stro simply not symmetric".

Is a Roadie with wider tyres still fas with slicks?

OK to leave the office without permi from home?

In HP, are memories reusable?

Incomplete borders around div

Do men feel threatened by intellige

Is it OK to sort /etc/ld.so.conf

(100%) 22:22

goohooh@goohooh-VirtualBox: ~/mysite

```
detected number of CPU cores: 1
current working directory: /home/goohooh/mysite
detected binary path: /usr/local/bin/uwsgi
!!! no internal routing support, rebuild with pcre support !!!
chdir() to /home/mysite
chdir(): No such file or directory [core/uwsgi.c line 2586]
chdir(): No such file or directory [core/uwsgi.c line 1608]
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$ sudo apt-get remove uwsgi
[sudo] password for goohooh:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  sqlite3
Use 'apt-get autoremove' to remove it.
다음 패키지를 지울 것입니다:
  uwsgi
0개 업그레이드, 0개 새로 설치, 1개 제거 및 239개 업그레이드 안 함.
이 작업 후 101 kB의 디스크 공간이 비워집니다.
계속 하시겠습니까? [Y/n] y
(데이터베이스 읽는 중 ...현재 181208개의 파일과 디렉터리가 설치되어 있습니다.)
Removing uwsgi (1.9.17.1-5build5) ...
 * Stopping app server(s) uwsgi                                    [ OK ]
(uwsgi-tutorial) goohooh@goohooh-VirtualBox:~/mysite$
```

????

넋을 놓고 타이핑을 하다가....

여기서부터 의욕을 잃었습니다. 내용을 전부 알고 이해하는 것이 아니라 추측하면서(영어공부 부족이지만..) 진행하다 보니 이걸 왜 하는 건지 이해가 안 갈 때도 있었습니다.

막상 진행상황을 나열해보니 한건 얼마 없는데 시간은 엄청 잡아 먹었습니다. 데이터베이스 연동은 손도 못댔습니다.

일단 진행한 부분만 제출합니다. 죄송합니다.