Setting up Apache 2.4 and web.py

Shaun Schreiber

1 Installation

1.1 Apache 2.4

Type the following commands into the terminal:

- `sudo apt-get install apache2-dev`
- `sudo apt-get install apache2-threaded-dev`
- `sudo apt-get install libapache2-mod-wsgi`
- `sudo apt-get install libapache2-mod-python`

1.2 web.py

Type the following commands into the terminal:

- `sudo apt-get update`
- `sudo apt-get install python-pip`
- `sudo pip install web.py`

2 Setting up basic project

2.1 Frame work

Type the following commands into the terminal:

- `sudo mkdir /var/www/webpy-app/<app name>`
- `sudo mkdir /var/www/webpy-app/<app name>/logs`
- Create two files in logs – access.log and error.log
- Create a python file in the `/var/www/webpy-app/<app name>` directory. Lets call it `main.py`
- `cd /etc/apache2/sites-available`

- Create file without an extension with the name same name as your project, lets call it `<project name>` . Note the project name could be the same as your `<app name>` but it does not have to be.

2.2 Configure Apache 2.4

Firstly connect the server IP to a name. Note that www.mysite.co.za refers to the server name. Next go to the `/etc` directory and open the file hosts. Add 127.0.0.1 mysite.co.za just below localhost. Note that I am using the localhost IP address as my server IP for this demonstration. Save the file – you will need to have super user privileges to do this. Navigate to `/etc/apache2/sites-enabled`, open the <project name> file and type the following:

```
<VirtualHost <server name> or IP:port>
     #ServerAdmin [your admin mail address]
     ServerName superfluous.imqs.co.za
     ServerAlias superfluous.imqs.co.za
     DocumentRoot /var/www/webpy-app/<app name>>
     DirectoryIndex <main python file name>.py
     WSGIScriptAlias /omicron /var/www/webpy-app/<app name>
        /<main python file name>.py/
     ErrorLog /var/www/webpy-app/<app name>/logs/error.log
     CustomLog /var/www/webpy-app/<app name>/logs/access.log combined
     addType text/html .py
<Files <main python file>.py>
        SetHandler wsgi-script
        Options ExecCGI FollowSymlinks
</Files>
<Directory />
        Order Allow,Deny
        Allow From All
        Options -Indexes
</Directory>
<Directory /var/www/webpy-app/<app name>>
        Options +ExecCGI +Indexes +MultiViews +FollowSymLinks
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
</VirtualHost>
```

Here is an example.

```
<VirtualHost *:80>
     #ServerAdmin [your admin mail address]
     ServerName omicron.imqs.co.za
     ServerAlias omicron.imqs.co.za
     DocumentRoot /home/ubuntu/Omicron/code/src
     DirectoryIndex webservice.py
     WSGIScriptAlias /omicron /home/ubuntu/Omicron/code/src/webservice.py/
     ErrorLog /home/ubuntu/Omicron/code/src/logs/error.log

     CustomLog /home/ubuntu/Omicron/code/src/logs/access.log combined
     addType text/html .py
     Alias /web /home/ubuntu/Omicron/code/src/web
 <Directory />
        Order Allow,Deny
        Allow From All
        Options -Indexes
    </Directory>
<Files webservice.py>
        SetHandler wsgi-script
        Options ExecCGI FollowSymlinks
</Files>
<Directory /home/ubuntu/Omicron/code/src>
        AddHandler server-parsed .js
        Options +ExecCGI +Indexes +MultiViews +FollowSymLinks
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
</VirtualHost>
```

You can test the configuration by typing the following command:

- `sudo apachectl configtest`
- `sudo a2ensite <project name>`

Only continue if the last line reads "`Syntax OK`". Navigate to `/var/www/webpy-app/<app name>/`, open `main.py` and type the following:

```
import sys
sys.path.append("/var/www/webpy-app/<app name>/")
import web
if app_path: # Apache
    os.chdir(app_path)
else: # CherryPy
    app_path = os.getcwd()
urls = (
    '/(.*)', 'hello'
)
# WARNING
# web.debug = True and autoreload = True
# can mess up your session: I've personally experienced it
```

```
web.debug = False # You may wish to place this in a config file
app = web.application(urls, globals(), autoreload=False)
application = app.wsgifunc() # needed for running with apache as wsgi. The reason is bec
class hello:
    def GET(self, name):
        if not name:
            name = 'World'
        return 'Hello, ' + name + '!'
if __name__ == "__main__":
    app.run()
```

Save the file and restart Apache by typing one of the following commands:

- `sudo service apache2 restart` **or**
- `sudo apachectl restart`

To test the server; open the browser, and type the following `<server name>/<project name>/hello` into your browser. If you are looking for more information you can visit the following links.

- `http://www.hyperink.com/blog/?p=13`
- `http://webpy.org/cookbook/mod_wsgi-apache-ubuntu`
- `http://webpy.org/cookbook/cgi-apache`
- `http://www.youtube.com/watch?v=831OahgMR9k`