

Installer un serveur web configuré pour python. (lighttpd+fastcgi+Flask)

Meilland jean claude

16 mars 2019

Avant de commencer je précise que vous avez un tutoriel pour Apache + PHP + Mysql sur raspbian-france

1 Installation

- **lighttpd** est un serveur léger donc idéal pour notre raspberry.
- **Flask** est un framwork pour réaliser un site internet en python.
- **Flup** est un module python pour *Web Server Gateway Interface* (WSGI)

Commençons par installer le serveur et flask.

```
sudo apt-get install lighttpd
sudo apt-get install python3-flask
sudo pip3 install flup
```

Les différents modules que l'on peut charger se trouvent dans le dossier `/etc/lighttpd/conf-available/`

```
ls /etc/lighttpd/conf-available/
```

Nous utiliseront les modules **10-fastcgi.conf**, **10-rewrite.conf** et **10-userdir.conf**.

```
sudo lighttpd-enable-mod fastcgi
sudo lighttpd-enable-mod rewrite
sudo lighttpd-enable-mod userdir
```

Les modules chargés se trouvent dans le dossier `/etc/lighttpd/conf-enabled/`

```
ls /etc/lighttpd/conf-enabled/
```

Il ne reste plus qu'à relancer **lighttpd**

```
sudo service lighttpd restart
```

2 Vérification

→ Vérifions que le **lighttpd** soit fonctionnel.

Ouvrez un navigateur à l'adresse `http://localhost/`. Vous devez voir cette page

Placeholder page
The owner of this web site has not put up any web pages yet. Please come back later.

You should replace this page with your own web pages as soon as possible.

Unless you changed its configuration, your new server is configured as follows:

- Configuration files can be found in `/etc/lighttpd`. Please read `/etc/lighttpd/conf-available/README` file.
- The DocumentRoot, which is the directory under which all your HTML files should exist, is set to `/var/www/html`.
- CGI scripts are looked for in `/usr/www/cgi-bin`, which is where Debian packages will place their scripts. You can enable cgi module by using command "**lighty-enable-mod cgi**".
- Log files are placed in `/var/log/lighttpd`, and will be rotated weekly. The frequency of rotation can be easily changed by editing `/etc/logrotate.d/lighttpd`.
- The default directory index is `index.html`, meaning that requests for a directory `/foo/bar/` will give the contents of the file `/var/www/foo/bar/index.html` if it exists (assuming that `/var/www` is your DocumentRoot).
- You can enable user directories by using command "**lighty-enable-mod userdir**".

About this page

This is a placeholder page installed by the Debian release of the **Lighttpd server package**.

This computer has installed the Debian GNU/Linux operating system, but it has nothing to do with the Debian Project. Please do not contact the Debian Project about it.

If you find a bug in this Lighttpd package, or in Lighttpd itself, please file a bug report on it. Instructions on doing this, and the list of known bugs of this package, can be found in the **Debian Bug Tracking System**.

→ Idem pour le module *userdir*

Ouvrez un navigateur à l'adresse `http://localhost/~pi/`, vous devez voir la même page.

Évidemment remplacez *pi* par votre *\$USER*.

Le module *userdir* permet d'écrire les programmes dans le dossier `/home/pi/public_html/`

→ Vérifions le module *Flask*. Commençons par tester un mini site *Pythoninfo* (comme *phpinfo*).

```
mkdir ~/public_html/pythoninfo
sudo pip3 install pyinfo
```

Créer le fichier `~/public_html/pythoninfo/pythoninfo.py`

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
from flask import Flask
app = Flask(__name__)


@app.route('/pyinfo')
@app.route('/')
def info():
    import pyinfo
    return pyinfo.info_as_html()

if __name__ == '__main__':
    app.run(debug=True)
```

On peut déjà tester ce fichier en local

```
python3 pythoninfo.py
```

Ouvrez votre navigateur à l'adresse `http://localhost:5000` Vous devez voir ceci

Python 3.5.3		
System		
Python version	3.5.3	
OS Version	Linux 4.9.59-v7+ (Debian 9.3)	
Executable	/usr/bin/python3	
Build Date	Jan 19 2017 14 :11 :04	
Compiler	GCC 6.3.0 20170124	
Python API	1013	

→ Configurez *lighhttpd* pour notre site internet *pythoninfo*.

Créer le fichier `~/public_html/pythoninfo/pythoninfo.fcgi`

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

from flup.server.fcgi import WSGIServer
from pythoninfo import app

if __name__ == '__main__':
    WSGIServer(app).run()
```

Rendez ce fichier exécutable.

```
chmod +x ~/public_html/pythoninfo/pythoninfo.fcgi
```

Ouvrez avec les droits root le fichier */etc/lighttpd/conf-enabled/10-fastcgi.conf*

```
sudo editor /etc/lighttpd/conf-enabled/10-fastcgi.conf
```

Ajouter au bas du fichier

```
# http://localhost/~pi/pythoninfo
fastcgi.server += ( "/~pi/pythoninfo" =>
    (
        "socket" => "/tmp/pythoninfo-user-fcgi.sock",
        "bin-path" => "/home/pi/public_html/pythoninfo/pythoninfo.fcgi",
        "check-local" => "disable",
        "max-procs" => 1
    )
)

url.rewrite-once += (
    "~/~pi/pythoninfo$" => "/~pi/pythoninfo/"
)
```

vérifiez que la syntaxe soit correcte avec la commande suivante :

```
lighttpd -t -f /etc/lighttpd/lighttpd.conf
```

Il faut relancer le service lighttpd.

```
sudo service lighttpd restart
```

Il ne reste plus qu'à vérifier. Ouvrez votre navigateur à l'adresse <http://localhost/~pi/pythoninfo/> ou à l'adresse <http://localhost/~pi/pythoninfo/pyinfo>

3 Un exemple de formulaire avec flask

3.1 Formulaire avec redirection sur une deuxième page

Vous pouvez télécharger directement les fichiers <http://megamaths.hd.free.fr/~pi/statics/testflask.zip>

→ On créer les dossiers puis on télécharge deux images pour le site.

```
mkdir -p ~/public_html/testflask/templates
mkdir ~/public_html/testflask/static
cd ~/public_html/testflask/static/
wget http://megamaths.hd.free.fr/~pi/statics/favicon.ico
wget http://megamaths.hd.free.fr/~pi/statics/fleur.png
```

→ Créer le fichier *~/public_html/testflask/testflask.py*

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
from flask import Flask, render_template, url_for, request
app = Flask(__name__) # Initialise l'application Flask
```

```

@app.route('/') # C'est un décorateur, on donne la route ici "/" l'adresse sera donc
                localhost:5000/
def page1():
    return render_template("formulaire1_page1.html", titre="Formulaire 1")

@app.route('/hello', methods=['POST'])
def hello():
    Nom=request.form['nom']
    Prenom=request.form['prenom']
    return render_template('formulaire1_page2.html', titre="Page 2", nom=Nom, prenom=
    Prenom)

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)

```

→ Créer le fichier ~/public_html/testflask/formulaire1_page1.html

```

<html>
<head>
<meta charset="utf-8" />
<title>{{ titre }}</title>
<!-- On importe notre frichier css -->
<link href="{{ url_for('static', filename='mon_style.css') }}" rel="stylesheet" type="
    text/css" />
<!-- Les images -->
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">
</head>

<body>
<header><h1>{{ titre }} </h1></header>
<section>
<div>
    <form method="post" action="{{ url_for('hello') }}">
    <label for="nom">Entrez votre nom:</label>
    <input type="text" name="nom" /><br />
    <label for="prenom">Entrez votre prénom:</label>
    <input type="text" name="prenom" /><br />
    <input type="submit" type="submit" value="Envoyer" />
    </form>
</div>
</section>
</body>
</html>

```

→ Créer le fichier ~/public_html/testflask/formulaire1_page2.html

```

<html>

```

```

<head>
<meta charset="utf-8" />
<title>{{ titre }}</title>
<!-- On importe notre frichier css -->
<link href="{{ url_for('static', filename='mon_style.css') }}" rel="stylesheet" type="
    text/css" />
<!-- Les images -->
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">
</head>

<body>
<header class="site-header"><h1>{{ titre }} </h1></header>
<section>
<div>
<p>bonjour <b><em>{{ prenom }} {{ nom }}</em></b></p>
<p><a href="{{ url_for('page1') }}" >Lien pour revenir sur la 1° page</a></p>
<!-- Attention le lien redirige sur la fonction python, pas sur le template html -->
</div>
</section>
</body>
</html>

```

→ N'oublions pas le fichier *public_html/testflask/static/mon_style.css*

```

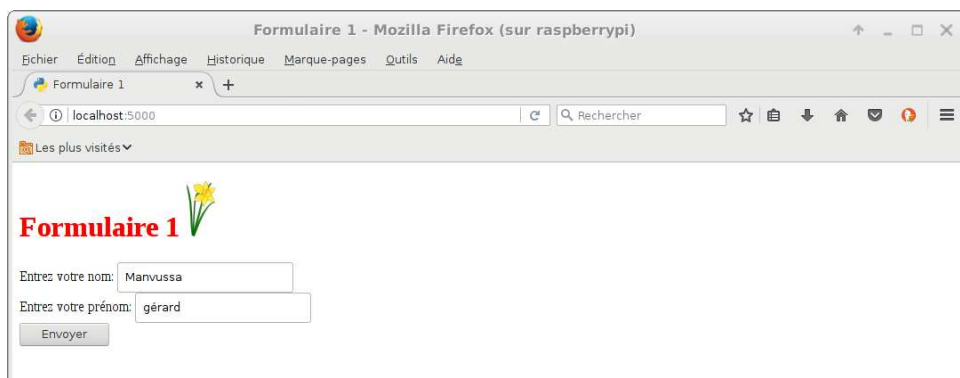
h1 {
    color: red;
}

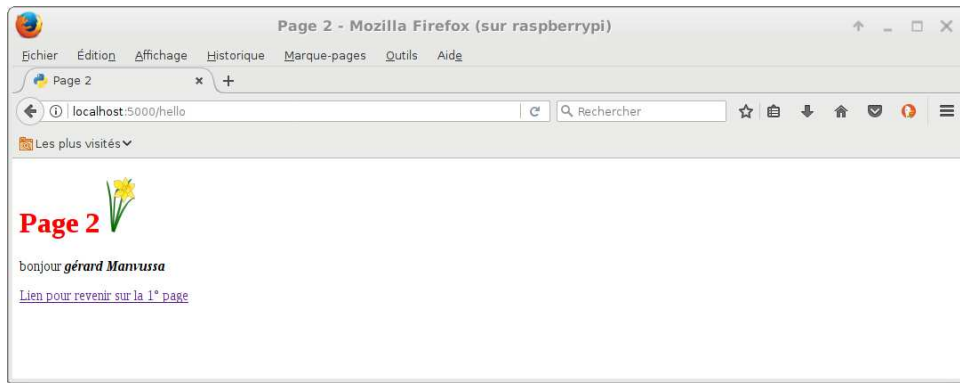
```

→ Exécuter le fichier python

```
python3 ~/public_html/testflask.py
```

Lancer votre navigateur à l'adresse <http://localhost:5000>





- Configurer *lighttpd* pour notre site internet *testflask*.
Cr  er le fichier `~/public_html/testflask/testflask.fcgi`

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
from flup.server.fcgi import WSGIServer
from testflask import app

if __name__ == '__main__':
    WSGIServer(app).run()
```

Rendez ce fichier ex  cutable.

```
chmod +x ~/public_html/pythoninfo/testflask.fcgi
```

Ouvrez avec les droits root le fichier `/etc/lighttpd/conf-enabled/10-fastcgi.conf`

```
sudo editor /etc/lighttpd/conf-enabled/10-fastcgi.conf
```

Ajouter au bas du fichier

```
# http://localhost/~pi/testflask
fastcgi.server += ( "~/pi/testflask" =>
    (
        "socket" => "/tmp/testflask-user-fcgi.sock",
        "bin-path" => "/home/pi/public_html/testflask/testflask.fcgi",
        "check-local" => "disable",
        "max-procs" => 1
    )
)

url.rewrite-once += (
    "~/pi/testflask$" => "/~pi/testflask/"
)
```

V  rifier la syntaxe et relancer le serveur *lighttpd*.

```
lighttpd -t -f /etc/lighttpd/lighttpd.conf
sudo service lighttpd restart
```

Il ne reste plus qu'   v  rifier. Ouvrez votre navigateur    l'adresse `http://localhost/~pi/testflask` ou    l'adresse `http://votrenomdedomaine/~pi/testflask`

3.2 Formulaire avec redirection sur la même page

→ Sauvegardons notre fichier python.

```
cp ~/public_html/testflask/testflask.py ~/public_html/testflask/testflask.old.py
```

→ Modifier le fichier `~/public_html/testflask/testflask.py`

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
from flask import Flask, render_template, url_for, request
app = Flask(__name__) # Initialise l'application Flask

@app.route('/', methods=['GET', 'POST']) # On doit indiquer que l'on utilise les deux
    méthodes
def page1():
    try:
        nom=request.form['nom']
    except:
        nom=''
    try:
        prenom=request.form['prenom']
    except:
        prenom=''
    methode=format(request.method)
    return render_template("formulaire2.html", titre='Formulaire 2',methode=methode,
        nom=nom,prenom=prenom)

if __name__ == '__main__':
    app.run(host='0.0.0.0',debug=True)
```

→ Créer le fichier `~/public_html/testflask/formulaire2.html`

```
<html>
<head>
<meta charset="utf-8" />
<title>{{ titre }}</title>
<!-- On importe notre frichier css -->
<link href="{{ url_for('static', filename='mon_style.css') }}" rel="stylesheet" type="
    text/css" />
<!-- Les images -->
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">
</head>

<body>
<header><h1>{{ titre }} </h1></header>
<section>
    <p>Cette page a été obtenue avec la méthode <b><em>{{ methode }}</em></b>.</p>
{% if request.method == 'GET' %}
<div>
    <form method="post" action="{{ url_for('page1') }}">
```

```

<label for="nom">Entrez votre nom:</label>
<input type="text" name="nom" /><br />
<label for="prenom">Entrez votre prénom:</label>
<input type="text" name="prenom" /><br />
<input type="submit" type="submit" value="Envoyer" />
</form>
</div>
{% else %}
    {% if nom != '' or prenom != '' %}
        <p>bonjour <b><em>{{ prenom }} {{ nom }}</em></b></p>
    {% endif %}
{% endif %}
</section>
</body>
</html>

```

→ Relancer *lighttpd*

```
sudo service lighttpd restart
```

→ Ouvrez votre navigateur à l'adresse <http://votrenomdedomaine/~pi/testflask>