****

**Configuring Apache and FastCGI Server**

|  |
| --- |
| **Dokumentation** |
|  |

Datum, Unterschrift (Verantwortlicher)

**Creating a *.fcgi* file**

First you need to create the FastCGI server file. Let’s call it *yourapplication.fcgi*:

#!/usr/bin/python

from flup.server.fcgi import WSGIServer

from yourapplication import app

if \_\_name\_\_ == '\_\_main\_\_':

WSGIServer(app).run()

This is enough for Apache to work, however nginx and older versions of lighttpd need a socket to be explicitly passed to communicate with the FastCGI server. For that to work you need to pass the path to the socket to the WSGIServer:

WSGIServer(application, bindAddress='/path/to/fcgi.sock').run()

The path has to be the exact same path you define in the server config.

Save the yourapplication.fcgi file somewhere you will find it again. It makes sense to have that in /var/www/yourapplication or something similar.

Make sure to set the executable bit on that file so that the servers can execute it:

$ chmod +x /var/www/yourapplication/yourapplication.fcgi

## Configuring Apache

The example above is good enough for a basic Apache deployment but your .fcgi file will appear in your application URL e.g. example.com/yourapplication.fcgi/news/. There are few ways to configure your application so that yourapplication.fcgi does not appear in the URL. A preferable way is to use the ScriptAlias and SetHandler configuration directives to route requests to the FastCGI server. The following example uses FastCgiServer to start 5 instances of the application which will handle all incoming requests:

LoadModule fastcgi\_module /usr/lib64/httpd/modules/mod\_fastcgi.so

FastCgiServer /var/www/html/yourapplication/app.fcgi -idle-timeout 300 -processes 5

<VirtualHost \*>

ServerName webapp1.mydomain.com

DocumentRoot /var/www/html/yourapplication

AddHandler fastcgi-script fcgi

ScriptAlias / /var/www/html/yourapplication/app.fcgi/

<Location />

SetHandler fastcgi-script

</Location>

</VirtualHost>

These processes will be managed by Apache. If you’re using a standalone FastCGI server, you can use the FastCgiExternalServer directive instead. Note that in the following the path is not real, it’s simply used as an identifier to other directives such as AliasMatch:

FastCgiServer /var/www/html/yourapplication -host 127.0.0.1:3000

If you cannot set ScriptAlias, for example on a shared web host, you can use WSGI middleware to remove yourapplication.fcgi from the URLs. Set .htaccess:

<IfModule mod\_fcgid.c>

AddHandler fcgid-script .fcgi

<Files ~ (\.fcgi)>

SetHandler fcgid-script

Options +FollowSymLinks +ExecCGI

</Files>

</IfModule>

<IfModule mod\_rewrite.c>

Options +FollowSymlinks

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^(.\*)$ yourapplication.fcgi/$1 [QSA,L]

</IfModule>

Set yourapplication.fcgi:

#!/usr/bin/python

#: optional path to your local python site-packages folder

import sys

sys.path.insert(0, '<your\_local\_path>/lib/python<your\_python\_version>/site-packages')

from flup.server.fcgi import WSGIServer

from yourapplication import app

class ScriptNameStripper(object):

def \_\_init\_\_(self, app):

self.app = app

def \_\_call\_\_(self, environ, start\_response):

environ['SCRIPT\_NAME'] = ''

return self.app(environ, start\_response)

app = ScriptNameStripper(app)

if \_\_name\_\_ == '\_\_main\_\_':

WSGIServer(app).run()