

## Anexo de Lighttpd

En este documento detallaremos las principales opciones de configuración del Lighttpd y qué hacen dichas opciones. El fichero de configuración es el lighttpd.conf que se encuentra en la carpeta de configuración de Lighttpd. En el proyecto hemos decidido poner una configuración totalmente funcional para que la primera puesta a punto sea sencilla. De hecho, hemos incorporado también una página web hecha por nosotros en lugar de que saliera el típico "Server is running".

El fichero de configuración que hemos incorporado al proyecto es el siguiente:

```
# lighttpd configuration file

#

# use it as a base for lighttpd 1.0.0 and above

#

# $Id: lighttpd.conf 8860 2008-08-05 10:56:16Z engy $

##### Options you really have to take care of
#####

## modules to load

# at least mod_access and mod_accesslog should be loaded

# all other module should only be loaded if really necessary

# - saves some time

# - saves memory

server.modules          = (

#                          "mod_rewrite",
```

```
#         "mod_redirect",
#         "mod_alias",
#         "mod_access",
#         "mod_cml",
#         "mod_trigger_b4_dl",
#         "mod_auth",
#         "mod_status",
#         "mod_setenv",
#         "mod_fastcgi",
#         "mod_proxy",
#         "mod_simple_vhost",
#         "mod_evhost",
#         "mod_userdir",
#         "mod_cgi",
#         "mod_compress",
#         "mod_ssi",
#         "mod_usertrack",
#         "mod_expire",
#         "mod_secdownload",
#         "mod_rrdtool",
#         "mod_accesslog" )

## a static document-root, for virtual-hosting take look at the
## server.virtual-* options
server.document-root    = "/opt/share/www/lighttpd/"
```

```
## where to send error-messages to
server.errorlog      = "/opt/var/log/lighttpd/error.log"

# files to check for if ../ is requested
index-file.names    = ( "index.php", "index.html",
                        "index.htm", "default.htm",
                        "lighttpd/index.html" )

## set the event-handler (read the performance section in the
manual)
server.event-handler = "poll" # needed on OS X

# mimetype mapping
mimetype.assign      = (
    ".pdf"           => "application/pdf",
    ".sig"           => "application/pgp-signature",
    ".spl"           => "application/futuresplash",
    ".class"         => "application/octet-stream",
    ".ps"            => "application/postscript",
    ".torrent"       => "application/x-bittorrent",
    ".dvi"           => "application/x-dvi",
    ".gz"            => "application/x-gzip",
    ".pac"           => "application/x-ns-proxy-autoconfig",
    ".swf"           => "application/x-shockwave-flash",
    ".tar.gz"        => "application/x-tgz",
    ".tgz"           => "application/x-tgz",
```

".tar" => "application/x-tar",  
".zip" => "application/zip",  
".mp3" => "audio/mpeg",  
".m3u" => "audio/x-mpegurl",  
".wma" => "audio/x-ms-wma",  
".wax" => "audio/x-ms-wax",  
".ogg" => "application/ogg",  
".wav" => "audio/x-wav",  
".gif" => "image/gif",  
".jpg" => "image/jpeg",  
".jpeg" => "image/jpeg",  
".png" => "image/png",  
".xbm" => "image/x-xbitmap",  
".xpm" => "image/x-xpixmap",  
".xwd" => "image/x-xwindowdump",  
".css" => "text/css",  
".html" => "text/html",  
".htm" => "text/html",  
".js" => "text/javascript",  
".asc" => "text/plain",  
".c" => "text/plain",  
".cpp" => "text/plain",  
".log" => "text/plain",  
".conf" => "text/plain",  
".text" => "text/plain",  
".txt" => "text/plain",

```
".dtd"      =>  "text/xml",
".xml"      =>  "text/xml",
".mpeg"     =>  "video/mpeg",
".mpg"      =>  "video/mpeg",
".mov"      =>  "video/quicktime",
".qt"       =>  "video/quicktime",
".avi"      =>  "video/x-msvideo",
".asf"      =>  "video/x-ms-asf",
".asx"      =>  "video/x-ms-asf",
".wmv"      =>  "video/x-ms-wmv",
".bz2"      =>  "application/x-bzip",
".tbz"      =>  "application/x-bzip-compressed-tar",
".tar.bz2"  =>  "application/x-bzip-compressed-tar"
)
```

```
# Use the "Content-Type" extended attribute to obtain mime type if
possible
```

```
#mimetype.use-xattr      = "enable"
```

```
## send a different Server: header
```

```
## be nice and keep it at lighttpd
```

```
# server.tag              = "lighttpd"
```

```
#### accesslog module
```

```
accesslog.filename        = "/opt/var/log/lighttpd/access.log"
```

```
## deny access the file-extensions
#
# ~ is for backupfiles from vi, emacs, joe, ...
# .inc is often used for code includes which should in general not be
part
# of the document-root
url.access-deny = ( "~", ".inc" )

$HTTP["url"] =~ "\.pdf$" {
    server.range-requests = "disable"
}

##
# which extensions should not be handle via static-file transfer
#
# .php, .pl, .fcgi are most often handled by mod_fastcgi or mod_cgi
static-file.exclude-extensions = ( ".fcgi", ".php", ".pl", ".py", ".rb" )

##### Options that are good to be but not necessary to be
changed #####

## bind to port (default: 80)
server.port = 80

## bind to localhost (default: all interfaces)
#server.bind = "grisu.home.kneschke.de"
```

```
## error-handler for status 404
#server.error-handler-404 = "/error-handler.html"
#server.error-handler-404 = "/error-handler.php"

## to help the rc.scripts
server.pid-file = "/var/run/lighttpd.pid"

##### virtual hosts
##
## If you want name-based virtual hosting add the next three
settings and load
## mod_simple_vhost
##
## document-root =
## virtual-server-root + virtual-server-default-host + virtual-
server-docroot
## or
## virtual-server-root + http-host + virtual-server-docroot
##
#simple-vhost.server-root = "/home/weigon/wwwroot/servers/"
#simple-vhost.default-host = "grisu.home.kneschke.de"
#simple-vhost.document-root = "/pages/"

##
```

```
## Format: <errorfile-prefix><status-code>.html
```

```
## -> ../status-404.html for 'File not found'
```

```
#server.errorfile-prefix =  
"/home/weigon/projects/lighttpd/doc/status-"
```

```
## virtual directory listings
```

```
#dir-listing.activate = "enable"
```

```
## enable debugging
```

```
#debug.log-request-header = "enable"
```

```
#debug.log-response-header = "enable"
```

```
#debug.log-request-handling = "enable"
```

```
#debug.log-file-not-found = "enable"
```

```
### only root can use these options
```

```
#
```

```
# chroot() to directory (default: no chroot() )
```

```
#server.chroot = "/"
```

```
## change uid to <uid> (default: don't care)
```

```
#server.username = "nobody"
```

```
## change gid to <gid> (default: don't care)
```

```
#server.groupname = "nobody"
```

```
#### compress module
```



```
#compress.cache-dir      = "/tmp/lighttpd/cache/compress/"
#compress.filetype       = ("text/plain", "text/html")
```

```
##### proxy module
```

```
## read proxy.txt for more info
```

```
#proxy.server            = ( ".php" =>
#                          ( "localhost" =>
#                            (
#                              "host" => "192.168.0.101",
#                              "port" => 80
#                            )
#                          )
#                        )
```

```
##### fastcgi module
```

```
## read fastcgi.txt for more info
```

```
#fastcgi.server          = ( ".php" =>
#                          ( "localhost" =>
#                            (
#                              "socket" => "/tmp/php-fcgi.sock",
#                              "bin-path" => "/opt/bin/php-fcgi",
#                              "bin-environment" => (
#                                "PHP_FCGI_MAX_REQUESTS" =>
#                                "4000"
#                              )
#                            )
#                          )
#                        )
```

```
#  
#  
  
##### CGI module  
#cgi.assign = ( ".pl" => "/opt/bin/perl",  
# ".cgi" => "/opt/bin/perl" )  
#  
  
##### SSL engine  
#ssl.engine = "enable"  
#ssl.pemfile = "server.pem"  
#ssl.ca-file = "ca.crt"  
#ssl.verifyclient.activate = "enable"  
#ssl.verifyclient.enforce = "enable"  
#ssl.verifyclient.depth = 1  
  
##### status module  
#status.status-url = "/server-status"  
#status.config-url = "/server-config"  
  
##### auth module  
## read authentication.txt for more info  
#auth.backend = "plain"  
#auth.backend.plain.userfile = "lighttpd.user"  
#auth.backend.plain.groupfile = "lighttpd.group"
```

```

#auth.backend.ldap.hostname = "localhost"
#auth.backend.ldap.base-dn = "dc=my-domain,dc=com"
#auth.backend.ldap.filter = "(uid=$)"

#auth.require = ( "/server-status" =>
#
#     "method" => "digest",
#     "realm" => "download archiv",
#     "require" => "user=jan"
#
#     ),
#
#     "/server-config" =>
#
#     (
#
#         "method" => "digest",
#         "realm" => "download archiv",
#         "require" => "valid-user"
#
#     )
#
#     )

#### url handling modules (rewrite, redirect, access)
#url.rewrite = ( "^/$" => "/server-status" )
#url.redirect = ( "^/wishlist/(.+)" =>
"http://www.123.org/$1" )

#### both rewrite/redirect support back reference to regex
conditional using %n
#$HTTP["host"] =~ "^www\.(*)" {
# url.redirect = ( "^/(*)" => "http://%1/$1" )
#}

```

```
#

# define a pattern for the host url finding

# %% => % sign

# %0 => domain name + tld

# %1 => tld

# %2 => domain name without tld

# %3 => subdomain 1 name

# %4 => subdomain 2 name

#

#evhost.path-pattern      = "/home/storage/dev/www/%3/htdocs/"

##### expire module

#expire.url               = ( "/buggy/" => "access 2 hours", "/asdhas/"
=> "access plus 1 seconds 2 minutes")

##### ssi

#ssi.extension            = ( ".shtml" )

##### rrdtool

#rrdtool.binary           = "/opt/bin/rrdtool"

#rrdtool.db-name          = "/var/www/lighttpd.rrd"

##### setenv

#setenv.add-request-header = ( "TRAV_ENV" =>
"mysql://user@host/db" )

#setenv.add-response-header = ( "X-Secret-Message" => "42" )
```

```
## for mod_trigger_b4_dl

# trigger-before-download.gdbm-filename =
"/home/weigon/testbase/trigger.db"

# trigger-before-download.memcache-hosts = ( "127.0.0.1:11211" )

# trigger-before-download.trigger-url = "^/trigger/"

# trigger-before-download.download-url = "^/download/"

# trigger-before-download.deny-url = "http://127.0.0.1/index.html"

# trigger-before-download.trigger-timeout = 10

## for mod_cml

## don't forget to add index.cml to server.indexfiles

# cml.extension          = ".cml"

# cml.memcache-hosts     = ( "127.0.0.1:11211" )

#### variable usage:

## variable name without "." is auto prefixed by "var." and becomes
"var.bar"

#bar = 1

#var.mystring = "foo"

## integer add

#bar += 1

## string concat, with integer cast as string, result: "www.foo1.com"

#server.name = "www." + mystring + var.bar + ".com"

## array merge

#index-file.names = (foo + ".php") + index-file.names
```

```
#index-file.names += (foo + ".php")
```

```
#### include
```

```
#include /etc/lighttpd/lighttpd-inc.conf
```

```
## same as above if you run: "lighttpd -f /etc/lighttpd/lighttpd.conf"
```

```
#include "lighttpd-inc.conf"
```

```
#### include_shell
```

```
#include_shell "echo var.a=1"
```

```
## the above is same as:
```

```
#var.a=1
```

```
# Configuration from other optware packages
```

```
include_shell "cat /opt/etc/lighttpd/conf.d/*.conf"
```

---

En la primera parte de la configuración del servidor, tenemos activados sólo dos módulos, el de acceso al servidor y el del LOG para ver las peticiones y los fallos que pudiera haber al arrancarlo.

Una de las partes más importantes de configuración es asignar la ruta donde se pondrá la web en sí, esta línea define la ubicación:

```
server.document-root = "/opt/share/www/lighttpd/"
```

En el script de instalación y puesta en marcha, hemos puesto la web en esa ubicación para que el servidor la interprete adecuadamente.

Otra parte importante es dónde ubicar los logs de errores del servidor, la ruta es esta:

```
server.errorlog = "/opt/var/log/lighttpd/error.log"
```

A continuación tenemos diferentes extensiones de archivos para su transferencia. Lo dejamos todo por defecto.

En nuestro servidor también hemos incorporado un log de acceso a él, la ruta donde se guarda es la misma donde se guarda el log de errores:

```
accesslog.filename      = "/opt/var/log/lighttpd/access.log"
```

Los servidores web por defecto siempre escuchan en el Puerto 80, por este motivo hemos cambiado el puerto de configuración vía web del router al 8080 ya que no podemos tener ambos servicios escuchando en el mismo puerto.

Si queremos cambiar el puerto, deberemos tocar esta línea:

```
server.port             = 80
```

Este servidor también nos permite fijarlo en una dirección IP, mostrar errores 404 o ejecutar scripts. Una opción muy interesante es la de hacer repositorios donde la descarga de archivos es vía http en lugar de FTP, para conseguir esto basta con activar la opción:

```
#dir-listing.activate   = "enable"
```

También podemos poner en nuestra página cifrado SSL, aunque si el certificado no está reconocido por una CA, nos saldrá el típico mensaje de los navegadores diciendo que no es de confianza.