

Abilitare diverse versioni di PHP in PLESK

Amministrando un web server con PLESK prima o poi arriva la richiesta di installare una seconda versione di PHP e di renderla disponibile ai clienti attraverso il pannello PLESK.

In questa guida l'installazione è stata fatta su una macchina CentOS 6.6 e Plesk 11.5.30:

Per prima cosa creiamo la directory dove potere scaricare il pacchetto PHP

```
cd /usr/local/src
# mkdir php562
# cd php562
wget http://it1.php.net/get/php-5.6.2.tar.gz/from/this/mirror
mv mirror php-5.6.2.tar.gz
tar -xvzf php-5.6.2.tar.gz
cd php-5.6.2
```

Siamo pronti per iniziare, configuriamo per la compilazione:

```
./configure '--with-libdir=lib64' '--cache-file=../config.cache' '--
prefix=/usr/local/php562-cgi' '--with-config-file-path=/usr/local/php562-
cgi/etc' '--disable-debug' '--with-pic' '--disable-rpath' '--enable-fastcgi'
'--with-bz2' '--with-curl' '--with-xpm-dir=/usr/local/php562-cgi' '--with-
png-dir=/usr/local/php562-cgi' '--enable-gd-native-ttf' '--without-gdbm' '--
with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr/local/php562-
cgi' '--with-openssl' '--with-pspell' '--with-pcre-regex' '--with-zlib' '--
enable-exif' '--enable-ftp' '--enable-sockets' '--enable-sysvsem' '--enable-
sysvshm' '--enable-sysvmsg' '--enable-wddx' '--with-kerberos' '--with-
unixODBC=/usr' '--enable-shmop' '--enable-calendar' '--without-sqlite3' '--
with-libxml-dir=/usr/local/php562-cgi' '--enable-pcntl' '--with-imap' '--
with-imap-ssl' '--enable-mbstring' '--enable-mbregex' '--with-gd' '--enable-
bcmath' '--with-xmlrpc' '--with-ldap' '--with-ldap-sasl' '--with-mysql=/usr'
'--with-mysqli' '--with-snmp' '--enable-soap' '--with-xsl' '--enable-
xmlreader' '--enable-xmlwriter' '--enable-pdo' '--with-pdo-mysql' '--with-
pdo-pgsql' '--with-pear=/usr/local/php562-cgi/pear' '--with-mcrypt' '--
enable-intl' '--without-pdo-sqlite' '--with-config-file-scan-
dir=/usr/local/php562-cgi/php.d' --enable-shared --enable-zip
```

Ora il classico make e poi make install, mi raccomando non lanciate make test

```
make
make install
```

Copiamo il php.ini sotto nella directory php562-cgi

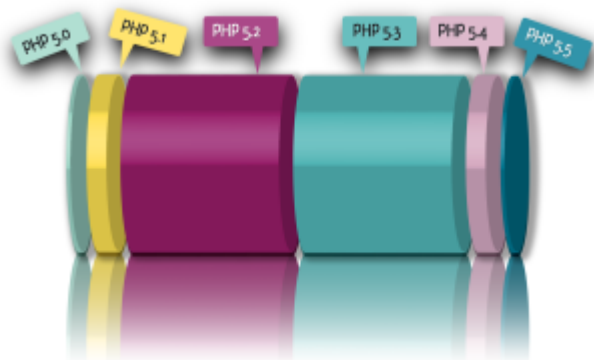
```
cp php.ini-development /usr/local/php562-cgi/php.ini
```

Ora non rimane che censire nel pannello PLESK la nuova versione di PHP

```
/usr/local/psa/bin/php_handler --add -displayname 5.6.2 -path  
/usr/local/php562-cgi/bin/php-cgi -phpini /usr/local/php562-cgi/php.ini -type  
fastcgi -id 5.6.2
```

Nel mio caso la prima installazione di PHP non è stata così liscia, ho dovuto installare alcuni pacchetti:

```
yum install bzip2-devel.x86_64 bzip2.x86_64  
yum install libjpeg*  
yum install libpng-devel  
yum install freetype  
yum install libXpm-devel  
yum install libgmp3-dev gmp.x86_64 gmp-devel.x86_64  
yum install openssl openssl-devel pam-devel  
yum install pam-devel  
yum install libicu-devel libc-client-devel.x86_64 libc-client.x86_64  
yum install libtomcrypt-devel.x86_64 libmcrypt-devel.x86_64 php-mcrypt.x86_64  
yum install unixODBC-devel  
yum install postgresql-devel postgresql-libs  
yum install pspell php-pspell.x86_64 aspell-devel net-snmp-devel libxslt-  
devel libxml2-devel pcre-devel tlib-devel.x86_64 libtidy-devel php-pecl-zip
```



[Symfony - 01 - Installazione e configurazione del framework](#)

Credo sia utile una mini guida su come installare e utilizzare il framework php Symfony2.

Innanzitutto preparatevi un ambiente LAMP, Linux + Apache + Mysql + Php

E prepariamoci un nuovo database mysql, che chiamiamo symfony , che dovremmo utilizzare per abilitare l'accesso securizzato.

Spostiamoci sotto la DocumentRoot di apache (se avete l'installazione di default, è /var/www/html)

Da qui, scarichiamo il composer, che serve per gestire le dipendenze del framework

```
curl -s https://getcomposer.org/installer | php
```

Ed ora scarichiamo il framework (ci vorrà qualche minuto).
Ho scelto la versione 2.3 che è supportata fino a Maggio 2016.

```
php composer.phar create-project symfony/framework-standard-edition symfony/  
"2.3.*"
```

Ora vi chiede le informazioni sul database creato per configurare il file
"##DOCUMENT_ROOT##/symfony/app/config/parameters.yml" (che potete configurare
in un secondo momento).

Ve le riassumo qui:

```
database_driver (pdo_mysql):  
database_host (127.0.0.1):  
database_port (null):  
database_name (symfony): symfony  
database_user (root): myuser  
database_password (null): mypassword  
mailer_transport (smtp):  
mailer_host (127.0.0.1):  
mailer_user (null):  
mailer_password (null):  
locale (en): it  
secret (ThisTokenIsNotSoSecretChangeIt): 2Cdq235nau
```

Il token potete generare una qualsiasi stringa.

Settiamo i permessi ai file scaricati:

```
chown -R apache.apache symfony
```

```
chmod -R 775 symfony
```

Ultimiamo l'installazione con:

```
cd symfony  
mv ../composer.phar .  
php composer.phar update  
chmod 777 app/cache/ app/logs/
```

Listiamo i file installati:

```
ls -l
```

E avremo in seguente output:

```
totale 116  
drwxrwxr-x 6 apache apache 4096 29 ago 11:44 app -> contiene log, cache,
```

configurazioni, risorse

```
drwxrwxr-x 2 apache apache 4096 29 ago 11:38 bin -> binari del framework
-rwxrwxr-x 1 apache apache 2085 8 lug 16:17 composer.json
-rwxrwxr-x 1 apache apache 56987 29 ago 11:38 composer.lock
-rwxrwxr-x 1 apache apache 1065 8 lug 16:17 LICENSE
-rwxrwxr-x 1 apache apache 5736 8 lug 16:17 README.md
drwxrwxr-x 3 apache apache 4096 8 lug 16:17 src -> file sorgenti del progetto
-rwxrwxr-x 1 apache apache 1308 8 lug 16:17 UPGRADE-2.2.md
-rwxrwxr-x 1 apache apache 1962 8 lug 16:17 UPGRADE-2.3.md
-rwxrwxr-x 1 apache apache 8499 8 lug 16:17 UPGRADE.md
drwxrwxr-x 13 apache apache 4096 29 ago 11:44 vendor -> framework
drwxrwxr-x 3 apache apache 4096 29 ago 11:44 web -> documentRoot del
framework
```

Ora se apriamo la seguente URL, modificando `##MYHOST##` con il nome del server, localhost se siamo in locale:

```
http://##MYHOST##/symfony/web/config.php
```

Si aprirà una pagina di verifica della configurazione, se non avete errori segnalati potete cliccare su

“Bypass configuration and go to the Welcome page”

Benvenuti nel vostro nuovo framework!

Se come errori segnalati, riscontrate il seguente:

```
Set the “date.timezone” setting in php.ini* (like Europe/Paris).
```

Aggiungete la stringa

```
date.timezone = Europe/Berlin
```

nel file php.ini (di default sotto /etc) e restartare apache.

Nel prossimo post vedremo come configurare l'accesso securizzato.

[Richiamare bash da php](#)

Gli script bash possono aiutarci ad interfacciarci al S.O. dal php

Per fare un esempio completo, create uno script che faccia una echo:

```
vi /home/roberto/script.sh
```

e dentro incollateci:

```
#!/bin/bash
```

```
echo "Ciao, sono la echo!"
```

E fate una pagina php con il seguente contenuto:

```
<?php
$result=shell_exec("/home/roberto/script.sh");
echo($result);
?>
```

Per sistemi unix/linux, se siete sotto Windows ovviamente i path saranno nel formato "C:\..."
