

---

# Complément Apache

## M2204-PHP , MySql et MariaDB

Patrice Gommery – Mars 2020



### Consignes Générales pour le TD

Pour réaliser les exercices de ce TD, vous disposez d'une machine virtuelle accessible à l'adresse : **172.16.N.250** (N étant votre n° de poste)

## Préambule : Etat des Lieux

Apache, MariaDB et PHP sont déjà installés sur votre machine.

Pour connaître les versions installées et actives :

<code>lsb_release -a</code>	: Version Debian
<code>apache2 -v</code>	: Version Apache2
<code>php -v</code>	: Version PHP (utilisée par la commande php)

---

mysql -v

: Version de Mysql ou MariaDB

REMARQUES IMPORTANTES :

1) Les versions installées ici sont celles fournies la Distribution Debian 9 , pas les dernières versions disponibles des logiciels. Nous verrons plus loin comment installer les dernières versions fournies par les concepteurs des logiciels.

# Partie I : PHP

Ce cours n'est pas un cours sur le langage PHP, la seule instruction php que devez connaître est **echo** . Nous utiliserons aussi la fonction **phpinfo()** qui permet d'afficher la configuration de php active sur le serveur ou le site. Le reste ne fait pas parti de ce cours.

Lorsque nous parlons de PHP dans ce cours, nous parlons donc de l'interpréteur PHP (le moteur) qui est chargé d'exécuter les instructions et les fonctions du langage. L'interpréteur peut être appelé de différentes façon :

- Soit directement avec la commande php (Exemple php -v)
- Soit à travers d'un module d'Apache ou de tout autre service supportant php.

## PHP et ligne de commandes

Lorsque vous tapez la commande php -v ,vous faites donc appel directement au moteur php qui est installé et configuré par défaut sur votre système. Vous pouvez donc directement utiliser l'interpréteur pour exécuter une instruction ou un script php, le résultat par défaut sera retourné à l'écran.

### EXERCICE 1:

Créez un fichier nommé **test.php** avec le contenu suivant :

```
<?php
echo "Hello World \n";
?>
```

Pour exécuter votre script, saisissez simplement : **php test.php**

Vous avez appelé directement l'interpréteur qui a exécuté la commande **echo** et renvoyé le résultat sur la sortie standard de notre système : l'écran.

Cet usage de PHP n'est pas le plus courant, mais il serait tout à fait possible de gérer tout notre système avec php, pour peu que le langage dispose des fonctions nécessaires et c'est le cas.

### EXERCICE 2:

Ajoutez les lignes suivantes à votre fichier **test.php** et exécutez-le

```
$output = shell_exec('ifconfig eth0');
echo $output."/n";
```

Sur le même principe, nous pourrions aussi créer des dossiers et des fichiers.

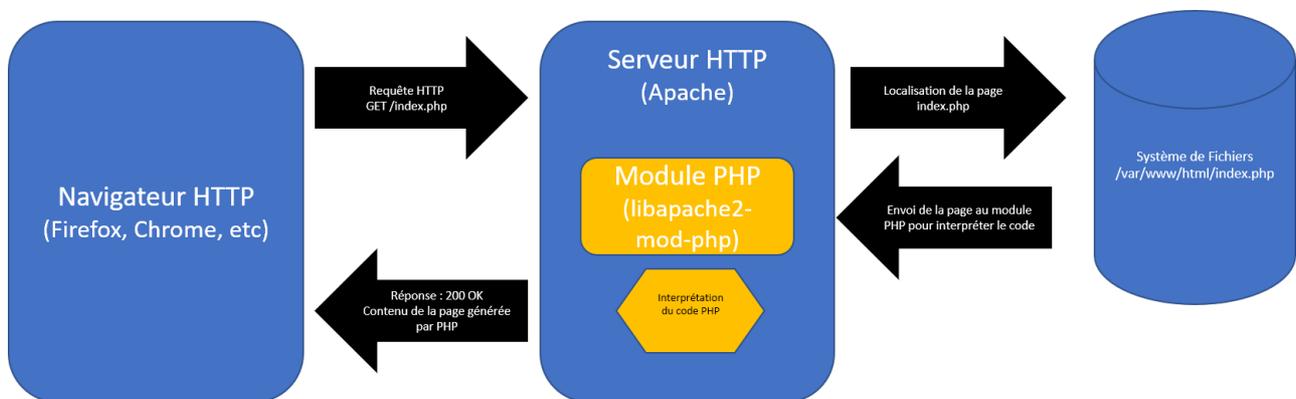
```
mkdir("/var/www/html/test", 0755);  
$file = fopen("/var/www/html/test/index.php", "x+");  
fputs($file,"<?php phpinfo(); ?>");  
fclose($file);
```

Pour vérifier, ouvrez votre navigateur avec l'adresse : **172.16.N.250/test/index.php**

La configuration complète de php devrait s'afficher.

## PHP et Apache

Lorsque vous créez une page web avec du code php, vous ne faites pas appel directement au moteur php, c'est apache qui s'en charge à travers un de ces modules. Le schéma de fonctionnement pour une page en php est le suivant.



1. Le navigateur envoie sa requête au serveur Web
2. Le serveur Web localise le fichier de la page demandée sur le système de fichiers
3. Si le fichier existe, il récupère son contenu et le transmet à l'interpréteur php
4. Le module php interprète le code et génère la page finale
5. Le serveur http envoie le résultat au navigateur

Comme vous le savez déjà, le navigateur ne reçoit aucun code en php, il ne sait interpréter que du html, du css ou du javascript.

Pour qu'Apache puisse communiquer avec le moteur PHP, il a donc besoin d'un module actif. Les modules disponibles sur votre serveur Apache se trouvent dans le dossier **/etc/apache2/mods-available**. Généralement, les modules sont constitués de deux fichiers (.load et .conf). Le premier charge le binaire du module, le second permet de le configurer.

### EXERCICE 3:

Localisez le fichier du module php présent sur votre machine et affichez son contenu. Comment interprétez-vous la section : **<IfModule mod\_userdir.c>** ?

Vous avez trouvé le nom du module php pour Apache, mais est-il vraiment actif ? Pour le savoir, regardez le contenu du dossier **/etc/apache2/mods-enabled**

Rappels :

---

Pour activer un module apache : **a2enmod NomDuModule**  
Pour désactiver un module : **a2dismod NomDuModule**

## La Bonne Version de PHP

Comme précisé au début de ce TD, la version de PHP installée sur votre système est celle livrée avec la distribution Debian que nous utilisons (`lsb_release -a`).

Ce n'est probablement pas la dernière version de PHP, puisque par principe, la version d'une release Debian reste la même pendant au moins un an et que ses composants restent les mêmes pour des questions de stabilité (ce qui fait la force de Debian).

Alors comment faire, si nous avons besoin d'une version plus récente de PHP pour exécuter une application particulière qui exploite de nouvelles fonctions. Ce cas est assez fréquent avec les Framework de développement comme par exemple Symfony que vous verrez au semestre 4.

Les dépôts Debian ne fournissant pas les packages que nous désirons, nous devons alors nous tourner vers des dépôts fournis par des développeurs de PHP ou de Debian. Un des plus connus (*que j'utilise depuis de nombreuses années sans aucun souci*) est celui de Sury

### EXERCICE 4:

Pour utiliser des dépôts supplémentaires sur votre distribution Debian la démarche est la suivante :

1) Installez le paquet suivant : **apt-transport-https**

REMARQUE : Les paquets `lsb_release`, `gnupg2` et `ca-certificates` sont aussi nécessaires mais déjà présents les machines virtuelles utilisées pour ce TD.

2) Téléchargez la clé **gpg** permettant de vérifier l'intégrité des paquets fournis par **Sury**. Votre serveur ne disposant pas d'accès direct à Internet, celle-ci est disponible sur le serveur FTP de la salle (**172.16.100.1**) dans le dossier **mmis2/Exemples** (*Login mmis2, PASSWORD*)

REMARQUE : Si vous disposiez d'un accès Internet, la commande serait :  
`wget -q https://packages.sury.org/php/apt.gpg`

3) Ajoutez la clé au système de gestion de paquets avec la commande :  
**apt-key add apt.gpg**

4) Ajoutez la déclaration du nouveau dépôt dans la liste des dépôts d'apt.

```
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main"  
| tee /etc/apt/sources.list.d/php.list
```

**ATTENTION** : La commande est à saisir sur une seule ligne !  
le `|` (pipe) s'obtient avec les touches `[Alt Gr]+6`

- 
- 5) Mettez à jour la liste des paquets avec la commande : **apt update**  
6) Installez les versions de PHP suivantes :

```
apt install php7.2  
apt install php5.6
```

Pour vérifier que les versions sont bien installées : **update-alternatives --list php**  
Logiquement, vous devriez voir 3 versions : php5.6, php7.0 et php7.2

Nous avons donc bien trois versions de PHP d'installées, mais laquelle est active ?

Pour le système : **php -v**

Pour Apache : ouvrez votre navigateur avec l'adresse : **172.16.N.250/test/index.php**

Avez-vous les mêmes versions ? Probablement Non, essayons de comprendre pourquoi.

Pour le système, c'est probablement la version la plus récente (7.4) qui est appelée par défaut. Normal, lorsque vous saisissez la commande php , vous appelez un raccourci qui pointe vers une version définie par défaut. Si par exemple, vous voulez fixer php5.6 comme version par défaut, utilisez la commande :

```
update-alternatives --set php /usr/bin/php5.6  
Pour vérifier: php -v
```

Pour utiliser une version particulière, utiliser le nom complet de l'exécutable de la version souhaitée . Par exemple **/usr/bin/php7.4 -v**

Pour Apache, c'est encore probablement la version 7.0 qui est active. Là aussi, rien d'anormal puisque l'installation d'un module n'implique pas forcément son activation. Vérifions d'abord quels sont les modules php disponibles dans notre installation d'Apache. Pour cela listez le contenu du dossier **/etc/apache2/mods-available** .

REMARQUE : Si un module n'est pas disponible, il est possible de l'installer directement . Les paquets se nomment libapache2-mod-phpX , ou X est la version de php désirée.

Les trois versions sont bien présentes, mais laquelle est active ?

Listez le contenu du dossier **/etc/apache2/mods-enabled**

Là une seule version est présente. Comment en activer une autre ?

Par exemple pour activer : **a2enmod php5.6**

Après avoir redémarrer Apache, vérifiez la version active avec votre navigateur et l'adresse : **172.16.N.250/test/index.php**

**ET LA C'EST LE DRAME** : Au mieux, Erreur 500 ou Impossible d'ouvrir la page. En fonction des tests effectués avant , il est même probablement impossible de relancer Apache. Que ce passe-t-il ? Simple : Les modules php entrent en conflit les uns avec les autres, impossible pour apache de savoir lequel exécuter.

Conclusion : IL NE PEUT EN RESTER QU'UN . Dans ce mode de fonctionnement, Apache n'accepte qu'un seul module PHP. Il faut donc désactiver les modules php en trop avec la commande **a2dismod**

### EXERCICE 5:

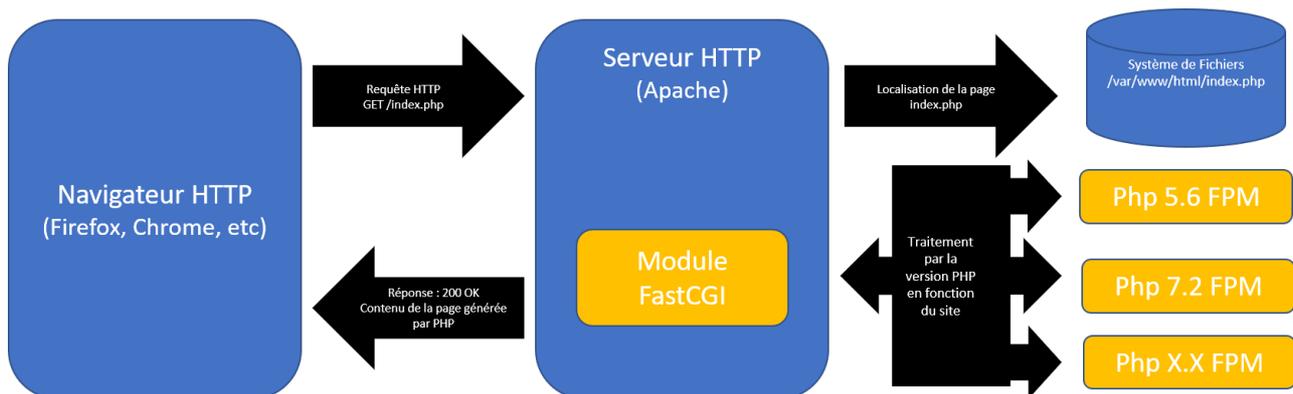
Ne laissez que la version php7.0 active pour Apache.  
Testez et validez avec votre page **index.php**

## Plusieurs Versions de PHP

Alors, est-il impossible d'avoir plusieurs sites avec des versions de PHP différentes ?  
Non, Apache possède un module permettant l'interfaçage avec des processus externes.

Il s'agit d'un module CGI (**Common Gateway Interface**) qui va donc permettre à Apache de passer des instructions à un autre processus et d'en récupérer le résultat. Pour que cela fonctionne, il faut juste que le processus externe puisse aussi être géré à travers cette interface. Dans notre cas, nous utiliserons le module `fast_cgi` d'Apache (`mod_fcgid`) et les versions FPM (**FastCGI Process Manager**) de PHP.

Le schéma de fonctionnement sera alors celui-ci :



1. Le navigateur envoie sa requête au serveur Web
2. Le serveur Web localise le fichier de la page demandée sur le système de fichiers
3. Si le fichier existe, il récupère son contenu et le transmet au module FastCGI
4. En fonction de la configuration du site, le code est interprété par une version de PHP
5. Le module FastCGI récupère le résultat
6. Le serveur http envoie le résultat au navigateur

On voit ici que les différentes versions de PHP sont des processus à part entière qui tournent en // . Elles peuvent donc elles-aussi être configurées différemment en fonction des besoins.

### EXERCICE 6 : Implémentation de Plusieurs Versions de PHP sur Apache

---

REMARQUE : Pour simuler plusieurs sites, nous utiliserons une fonctionnalité d'Apache que nous n'avons pas exploitée pendant les TP : La gestion des ports. Nous utiliserons donc toujours la même adresse : 172.16.N.250 mais associée à des ports différents, ce qui donnera

- 172.16.N.250:8056 pour le site en php 5.6
- 172.16.N.250:8072 Pour le site en php 7.2

1) Installez les paquets nécessaires :

**php5.6-fpm, php7.2-fpm** pour PHP  
**libapache2-mod-fcgid** pour Apache

2) Activez les modules Apache suivants :

**actions, fcgid, alias, proxy\_fcgi**

3) Ajoutons les nouveaux ports à Apache. Pour cela , éditez le fichier **/etc/apache2/ports.conf** et ajoutez les lignes **Listen 8056** et **Listen 8072**

4) Créez les dossiers pour les sites :

**/var/www/php56**  
**/var/www/php72**

5) Recopiez votre page **index.php** dans les deux dossiers.

6) **Créez un fichier de configuration pour chacun des sites.**

**Exemple pour le site en php 5.6 :**

```
<VirtualHost *:8056>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/php56
    DirectoryIndex index.php

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/php56>
        Options -Indexes +FollowSymLinks +MultiViews
        AllowOverride All
        Require all granted
    </Directory>

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/var/run/php/php5.6-fpm.sock|fcgi://localhost"
    </FilesMatch>

</VirtualHost>
```

7) Testez vos sites en utilisant les adresses :

**172.16.N.250:8056** pour le site en php 5.6  
**172.16.N.250:8072** pour le site en php 7.2

---

Normalement, vous devriez constater que phpinfo vous retourne des versions différentes de PHP. La page vous permet aussi de voir , quel fichier de configuration (php.ini) est utilisée par chacune des versions.

## Configuration et Modules de PHP

Comme vous pouvez le constater avec phpinfo , vos versions de PHP ne sont pas toutes configurées pareilles. Outre le fait que le fichier de configuration est différent :

<code>/etc/php/5.6/apache2/php.ini</code>	pour php 5.6
<code>/etc/php/7.2/apache2/php.ini</code>	pour php 7.2

REMARQUE : Le contenu du fichier php.ini sera traité pendant le prochain TP

L'autre aspect important de la configuration porte sur le nombre de modules installés et disponibles pour chacune des versions. En effet, si la version de base (7.0) a été installé avec la plupart des modules courants de php, il n'en est rien pour les versions que vous avez ajoutées par la suite. Pour s'en convaincre il suffit de lister le contenu des dossiers mods-available de chacune des versions.

```
/etc/php/5.6/mods-available  
/etc/php/7.0/mods-available  
/etc/php/7.2/mods-available
```

Par exemple, un module particulièrement important est absent des versions 5.6 et 7.2 : **mysqli** . Le module qui permet à php de dialoguer avec une base de données mysql ou mariadb. C'est aussi le cas des modules **xml**, **mbstring** ou **curl** qui sont souvent utilisés par les CMS comme Wordpress, les frameworks de développement comme Symfony ou tout simplement **phpMyAdmin** qui vous permet de gérer vos bases de données.

Chaque version ayant ses propres modules, il faut donc les installer manuellement en fonction des besoins.

Complétez votre installation en installant les modules pour chacune des versions :

```
php5.6-mysqli, php5.6-xml, php5.6-mbstring, php5.6-curl  
php7.2-mysqli, php7.2-xml, php7.2-mbstring, php7.2-curl
```

Cette liste n'est bien sûr pas exhaustive . N'hésitez pas à vos référer à la documentation php ou de votre application pour savoir quel module vous manque.

### EXERCICE 7:

Installez php7.4 sur votre Machine Virtuelle et configurez un site web fonctionnant avec cette version et accessible sur le port 8074.

