

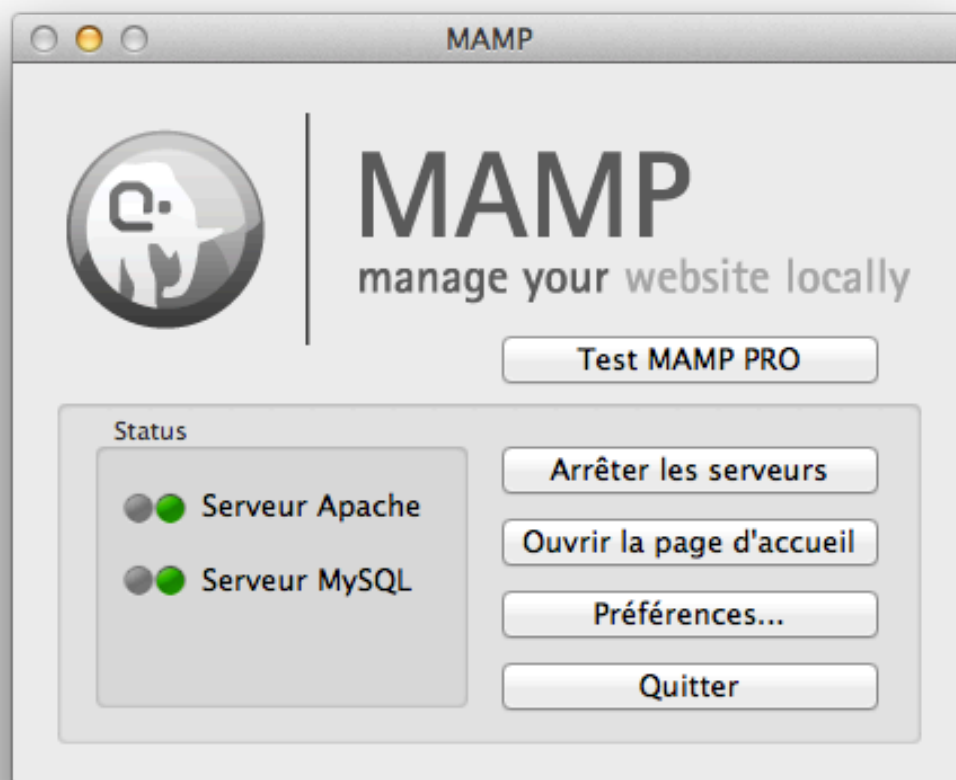
Configurer MAMP (pour Mac)

Quelques réglages utiles de base

Installation

L'installation est très facile. Après avoir téléchargé l'archive sur le site, il suffit de double-cliquer dessus...

A l'ouverture de MAMP (pour Mac-Apache-MySQL-PHP), on tombe sur cette fenêtre là :



Si tout se passe bien les leds de gauche doivent passer au vert. Si ce n'est pas le cas, c'est qu'un problème de configuration semble planter un des serveurs (ou les 2). Une réinstallation peut s'avérer nécessaire.

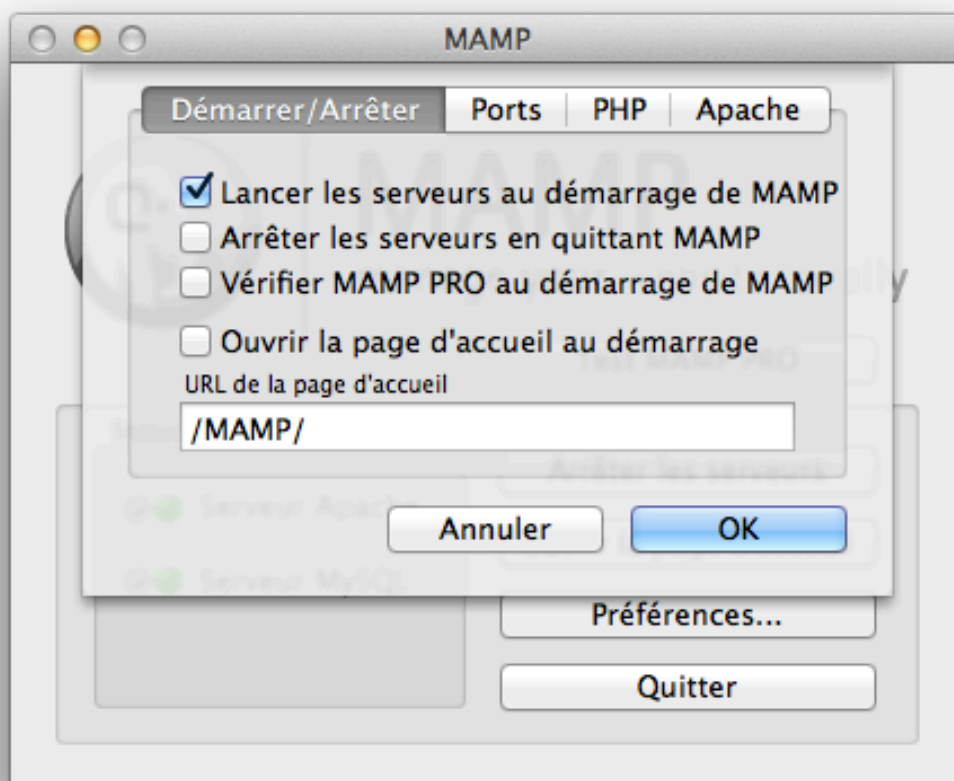
MAMP Pro (payant) permet d'effectuer plus simplement certains réglages, notamment la création de VirtualHost. Mais il n'est en aucun cas un passage obligé.

Une fois MAMP démarré, il est bon de faire un tour dans les préférences.

Réglages des préférences

Démarrer / Arrêter

Le premier onglet des préférences permet de régler certaines option concernant directement le fonctionnement de MAMP.



En général, par défaut, MAMP coupe les serveurs quand on le quitte. C'est la première case que je décoche aussitôt l'installation terminée.

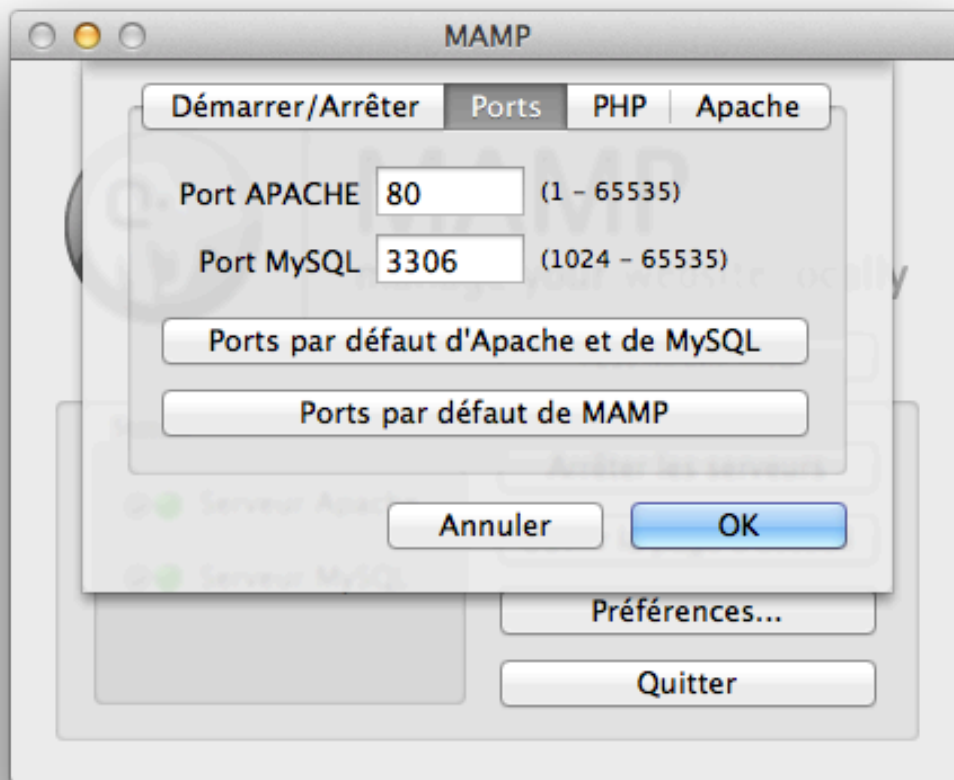
Bref, au lancement de MAMP, on le laisse lancer les serveurs, mais par contre, en le quittant on lui demande de ne pas les couper. Ça évite d'avoir une appli en pus dans le dock, et qui ne sert pas à grand chose. Au pire, le widget sur le dashboard permettra de relancer les serveurs si besoin.

La vérification de MAMP Pro est bien entendu à désactiver si vous ne voulez pas qu'il vous rappelle continuellement de passer à la caisse...

Enfin l'ouverture de la page d'accueil de MAMP au lancement est un choix personnel. Chez moi c'est désactivé.

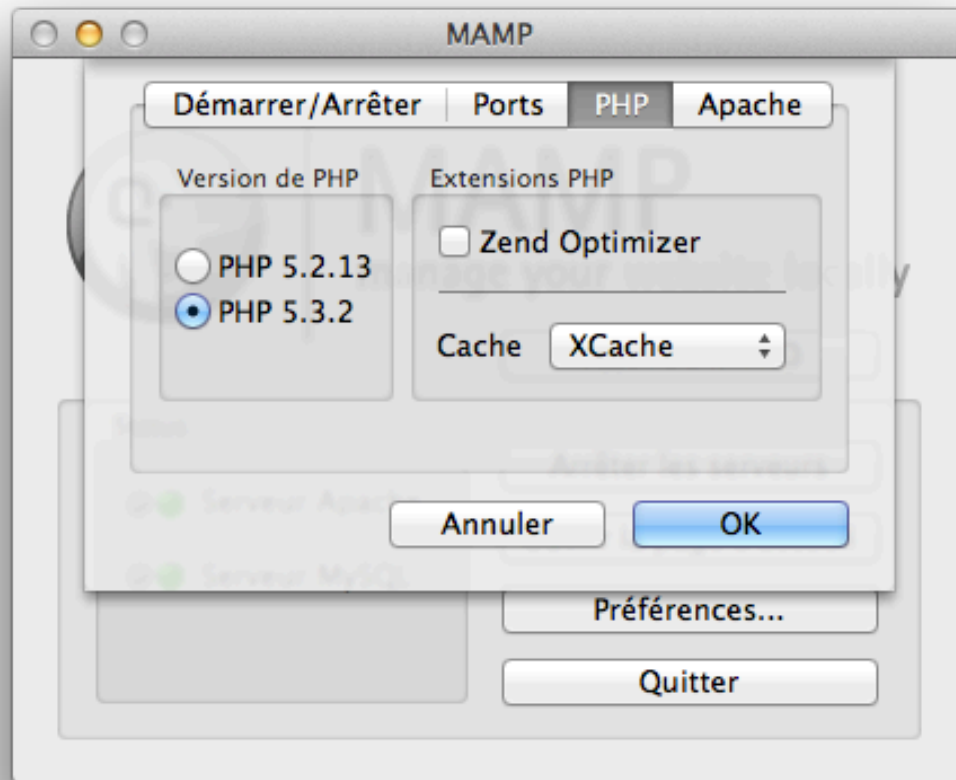
Réglages des ports

Les ports par défaut de MAMP sont tout à fait conservables. J'ai opté de mon côté pour utiliser les ports standards de Apache et MySQL, mais une protection sur MacOS X vous demandera votre mot de passe admin lors du lancement d'Apache. C'est une sécurité pour tous les ports inférieurs à 1000 (de mémoire pour cette valeur).



Réglages de PHP

Soyons clair, les options disponibles sont assez légères concernant PHP. Pour mieux le configurer il faudra aller éditer directement le fichier `php.ini`.

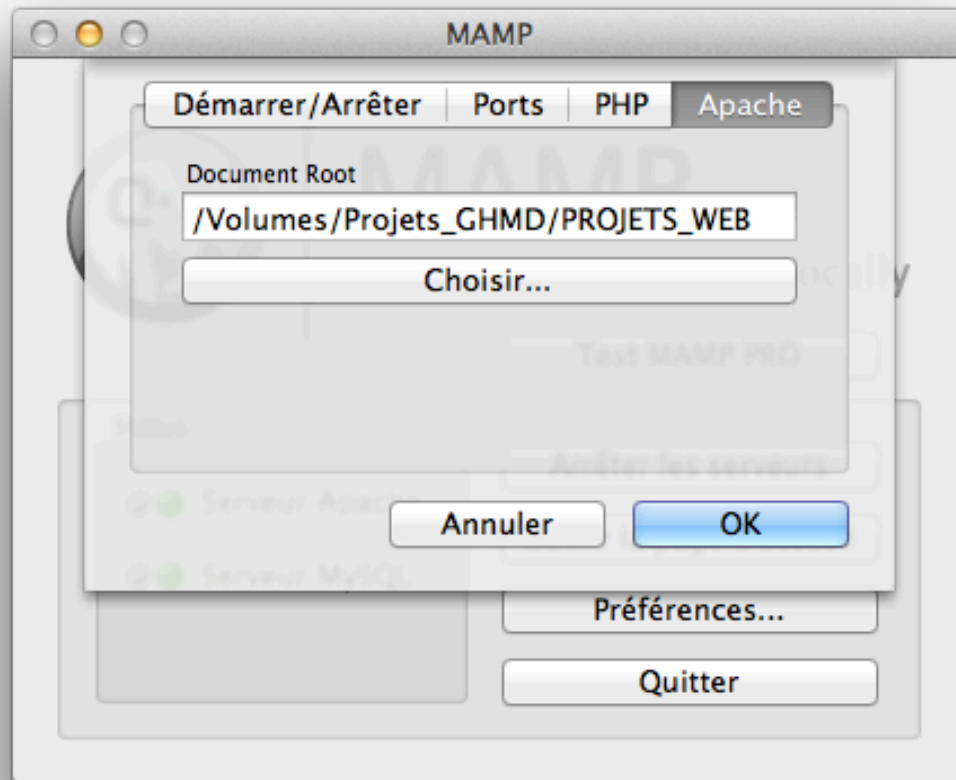


Ici donc, vous pouvez choisir la version de PHP à utiliser. Tout changement nécessitera de relancer les serveurs.

Concernant les caches, ceux-ci sont particulièrement tenaces voire même presque handicapant pour du développement. Nous allons aller les désactiver directement dans le fichier `php.ini`.

Réglages d'apache

De même que pour PHP, les réglages sont ici particulièrement limités. En fait, ils ne se bornent qu'à la définition du Document Root.



Par défaut, il s'agit du répertoire « htdocs » présent dans le dossier de MAMP. Cependant, il peut être judicieux de le changer pour un autre, afin de simplifier les sauvegardes ou d'y accéder plus simplement.

Dans mon cas, j'ai placé mon Document Root sur un disque externe, me permettant ainsi de travailler plus facilement sur différents ordinateurs.

Pour changer le Document Root, cliquez simplement sur le bouton « Choisir... » et laissez-vous guider.

Il faudra ensuite relancer les serveurs pour valider la prise en compte.

Astuces rendant la vie plus cool

En éditant directement les fichiers de configuration, vous pourrez activer/désactiver certaines options non disponibles dans les préférences de MAMP.

Le fichier `php.ini`

Ce fichier concerne les réglages propres à PHP.

Je ne vais pas lister ici tous les réglages possibles, mais certains peuvent être gênant dans un environnement de développement avec les valeurs par défaut.

Pour modifier ce fichier, je vous recommande l'utilisation d'un éditeur de code, même si TextEdit peut le faire, il est possible qu'il insère des caractères invisibles (des cafards) qui pourraient rendre le fichier inutilisable (et éventuellement faire planter Apache aussi).

Le fichier `php.ini` se trouve dans le dossier MAMP > bin > php > php[version choisie dans les préf] > conf > `php.ini`.

Par exemple : MAMP > bin > php > php5.5.3 > conf > `php.ini`

Dans ce fichier, les lignes commençant par « ; » sont des commentaires.

display_errors

La première valeur à modifier est certainement celle concernant l'affichage des erreurs dans vos pages. Très pratique pour le débogage.

Faites donc une recherche sur le terme « `display_errors` » et trouver la ligne :

```
display_errors = Off
```

(Aux alentours de la ligne 277)

Changez « Off » pour « On » afin de permettre l'affichage des erreurs.

error_reporting

Un peu plus haut que `display_errors` (ligne 270), vous aurez la propriété « `error_reporting` » qui définit quelles sont les erreurs à afficher. Le manuel est dans les commentaires juste au dessus, mais si ce n'est pas déjà le cas, optez pour la valeur « `E_ALL` ». Ce qui affichera aussi bien les erreurs fatales que les avertissements.

register_globals

Recherchez ensuite le terme « `register_globals` ».

Un fois la ligne trouvée (environ ligne 365), changez la valeur pour « Off » si ce n'est pas déjà le cas. En effet, mettre cette valeur à Off vous obligera à récupérer vos valeurs en passant par les variables super globales (`$_POST`, `$_GET`, etc.) et donc à coder plus proprement, tout en sécurisant le script et en évitant d'écraser involontairement des variables.

C'est normalement la valeur par défaut aujourd'hui sur PHP.

Si vous devez utiliser des scripts qui nécessitent de passer cette propriété à « On », vous saurez maintenant où la trouver !

post_max_size

Un peu en dessous des `register_globals`, « `post_max_size` » (ligne 373) permet de définir la valeur maximale en KB, MB, ... de ce qui peut être envoyé via un formulaire (texte et fichiers compris). 8M = 8Mo ; 200K = 200Ko ; ...

En général, je recommande de se caler sur les valeurs données par le `phpinfo` de son hébergeur pour éviter des surprises.

upload_max_filesize

Plus bas (ligne 481 environ), la propriété « `upload_max_filesize` » permet de définir la taille maximale des fichiers transférables par formulaire (avec la méthode POST bien sûr).

De même que pour `post_max_size`, je vous recommande de vous caler sur votre hébergeur.

memory_limit

Bien plus haut (ligne 232 environ), la propriété « `memory_limit` » permet de définir la limite mémoire utilisable par le script. C'est un garde fou, mais elle est bien souvent un chouilla trop basse par défaut.

Tout comme les 2 propriétés précédentes, je vous recommande de vous caler sur votre hébergeur.

Désactivation des caches

Les caches sont bien pratique en production car ils permettent l'économie de ressources.

Cependant, en phase de développement, on est très souvent en train de modifier et tester les fichiers. Les caches gardant quelques minutes en mémoire les fichiers, les modifications n'apparaissent pas immédiatement.

Il vaut donc mieux les désactiver pour éviter de belles prises de tête.

Pour ce faire :

- Cherchez « `opcache` ».
- Lignes 1086 à 1092 : Commentez les lignes (elles contiennent « `opcache` »)

En principe les autres propriétés peuvent rester par défaut pour la grande majorité du travail à faire.

Pensez à enregistrer et à redémarrer les serveurs.

Le fichier `httpd.conf`

htaccess

Ce fichier, et ses camarades, gère la configuration du serveur Apache. A priori, il y a peu de chose à faire dedans.

Ceci dit, une des modifications que je fais quasi-systématiquement est de changer le nom des fichiers `htaccess`.

Ces fichiers permettent d'utiliser l'URL Rewriting sur son site, mais comme leur nom commence par un point, ils ont tendance à devenir invisible sur MacOS X et du coup de ne pas être pris en compte.

L'astuce réside simplement à régler le serveur Apache pour qu'il cherche un autre fichier à la place. Personnellement, j'ai repris l'idée d'un de mes hébergeurs, consistant à utiliser le nom « `htaccess.fi` » plutôt que « `.htaccess` ».

Le réglage se fait dans le fichier `httpd.conf` qui se trouve dans `MAMP > conf > Apache > httpd.conf`

Chercher le terme « AccessFileName » (ligne 462 environ).
Changez « AccessFileName .htaccess » pour « AccessFileName htaccess.fi »

DirectoryIndex

Le DirectoryIndex permet de définir quel fichier sera appelé en priorité à l'appel d'un dossier. Par défaut se sera index.html. On peut en définir d'autres à la suite en les séparant par un espace et dans l'ordre de priorité souhaité.

Ex : DirectoryIndex index.html index.php index.htm

Vous trouverez se réglages juste avant AccessFileName, aux alentours de la ligne 455.