

# Beste de savoir

WAMP : développer avec PHP / AJAX /  
HTML sous Windows

---

13 juillet 2022



# Table des matières

<b>Introduction</b>	<b>3</b>
<b>I. L'environnement</b>	<b>5</b>
<b>Introduction</b>	<b>6</b>
<b>I.1. Vous avez dit serveur ?</b>	<b>7</b>
Introduction . . . . .	7
I.1.1. Le rôle du serveur . . . . .	7
I.1.2. Pour discuter avec un serveur : le protocole HTTP . . . . .	8
I.1.2.1. Une autre raison encore : le cas d'AJAX <sup>2</sup> . . . . .	9
Conclusion . . . . .	9
<b>I.2. Avant d'installer...</b>	<b>10</b>
Introduction . . . . .	10
I.2.1. IIS . . . . .	10
I.2.2. Skype . . . . .	11
I.2.3. SQL Server Reporting Services . . . . .	11
I.2.4. Vérifier si un autre programme utilise le port 80 . . . . .	11
I.2.4.1. Prévenir vaut mieux que guérir . . . . .	12
I.2.5. Désactiver les services plutôt que les désinstaller . . . . .	12
Conclusion . . . . .	12
<b>I.3. WAMP</b>	<b>13</b>
Introduction . . . . .	13
I.3.1. Sauvegarder, sauvegarder, sauvegarder ! . . . . .	14
I.3.1.1. Sauvegarder les scripts . . . . .	14
I.3.1.2. Sauvegarde des bases de données . . . . .	14
I.3.1.3. Sauvegarder la configuration . . . . .	15
I.3.2. Mise à jour de WAMP . . . . .	18
I.3.2.1. Mises à jour majeures . . . . .	18
I.3.2.2. Mises à jour mineures . . . . .	19
I.3.2.3. Patches . . . . .	20
I.3.3. Installation . . . . .	20
I.3.3.1. Est-ce que les services Apache et SGBD sont bien installés ? . . . . .	23
Conclusion . . . . .	24
<b>Conclusion</b>	<b>25</b>

<b>II. Utiliser son serveur</b>	<b>26</b>
<b>Introduction</b>	<b>27</b>
<b>II.1. Dans la vie de tous les jours...</b>	<b>28</b>
Introduction . . . . .	28
II.1.1. Paramétrer les composants . . . . .	28
II.1.1.1. Paramétrer Apache . . . . .	28
II.1.1.2. Paramétrer PHP . . . . .	29
II.1.1.3. Paramétrer votre SGBD . . . . .	30
II.1.2. Organiser vos projets . . . . .	31
II.1.2.1. Le dossier <code>www</code> . . . . .	31
II.1.2.2. Corriger les liens vers vos projets . . . . .	32
II.1.3. Les <code>&lt;VirtualHost&gt;</code> . . . . .	34
II.1.3.1. L'intérêt des hôtes virtuels . . . . .	34
II.1.3.2. Edition du fichier <code>hosts</code> de Windows . . . . .	36
II.1.3.3. Définition des hôtes virtuels . . . . .	37
II.1.3.4. Conserver l'accès à <code>http://localhost/</code> sur certaines versions de WAMP . . . . .	40
II.1.4. PHP en ligne de commandes . . . . .	41
II.1.5. Accéder à votre serveur depuis une machine distante . . . . .	43
II.1.5.1. S'ouvrir au monde . . . . .	44
II.1.5.2. Rediriger les gens chez vous . . . . .	44
II.1.5.3. Quelques obstacles possibles . . . . .	48
Conclusion . . . . .	49
Contenu masqué . . . . .	49
<b>II.2. ... ou pour des besoins spécifiques</b>	<b>51</b>
Introduction . . . . .	51
II.2.1. WAMP, ses modules, et Microsoft Visual C++ . . . . .	51
II.2.2. Plusieurs versions de PHP . . . . .	53
II.2.2.1. Télécharger les binaires pour Windows . . . . .	53
II.2.2.2. Installer la nouvelle version de PHP . . . . .	54
II.2.2.3. Eviter les incompatibilités avec ICU sous WAMP 2.5 (de 2.5.0 à 2.5.4 compris) . . . . .	56
II.2.2.4. Mettre à jour vers une révision plus récente . . . . .	58
II.2.3. Plusieurs versions du SGBD . . . . .	59
II.2.3.1. Installer la nouvelle version . . . . .	60
II.2.3.2. Partager les bases de données entre plusieurs versions . . . . .	61
II.2.4. Générer des certificats SSL/TLS et activer HTTPS . . . . .	63
II.2.4.1. Générer un certificat auto-signé . . . . .	63
Conclusion . . . . .	67
<b>Conclusion</b>	<b>68</b>

# **Introduction**

## *Introduction*

Comme beaucoup avant vous, vous avez lu des choses sur Internet, et vous vous êtes dit «C'est décidé, j'apprends le PHP !».

Vous avez alors commencé à regarder ce qu'il vous serait nécessaire, et vous avez compris qu'il vous faudrait un serveur, il a été fait mention d'Apache, de MySQL, et là, vous sentez votre enthousiasme diminuer...

Peut-être aurez-vous tenté d'installer diverses choses séparément, et vous avez abandonné parce que ça ne fonctionnait pas...

Vous êtes au bon endroit pour que votre moral remonte, parce que nous allons parler d'une solution "tout-en-un" qui répondra à vos besoins : **WAMP**.

Le logo utilisé pour le tutoriel est sous licence [GNU General Public License V2](#) [↗](#) ou plus récent.

# **Première partie**

## **L'environnement**

# Introduction

Ici, nous allons vous expliquer diverses choses à propos des environnements de développement sous Windows, puis nous allons vous expliquer comment en installer un tout en évitant certains pièges qui empêchent l'installation ou le fonctionnement corrects.

*i*

Quand nous parlons d'environnement de développement, nous ne parlons pas des éditeurs de code 🍊



# I.1. Vous avez dit serveur ?

## Introduction



Mais il n'y aura que moi et moi pour mes développements, je n'ai pas de serveur, j'en ai besoin ?

Oui, il vous en faudra un, et nous allons vous expliquer pourquoi 🍊

### I.1.1. Le rôle du serveur

Si vous vous engagez dans le développement web en général, sachez que sur Internet, il y a toujours au minimum deux intervenants:

- le **client**, c'est vous 🍊
- le **serveur**, qui est l'ordinateur où est enregistrée (on dira plutôt **hébergée**) la page que vous demandez

Comme au restaurant, vous faites votre commande pour la cuisine: il faut donc être au minimum deux<sup>1</sup>. Vous comprenez maintenant pourquoi c'est nécessaire. Et c'est donc ce que nous vous proposons de vous aider à mettre en place.

Cela peut vous paraître étrange, surtout parce que vous n'avez pas besoin d'un serveur si vous ne faites que du **HTML** avec du **CSS**. En fait, c'est parce que votre navigateur comprend les langages suivants:

- **HTML**, avec ses variantes **XHTML** et ses différentes versions
- **CSS**, qui permet de spécifier la mise en page de documents **HTML**
- JavaScript, qui permet entre autres les animations et le fait que la page ne se recharge pas complètement quand vous cliquez sur **+1** sur les forums de ce site par exemple
- **XML**, un format de données qui a inspiré **HTML**

Vous connaissez probablement les deux premiers, et avez entendu parler des autres. Ce qu'il faut retenir ici, c'est que le **PHP n'est pas compris par un navigateur**.



Mais si mon navigateur ne comprend pas le PHP, comment peut-il afficher des pages qui ont été écrites avec ? D'ailleurs, il affiche bien les pages qui ont une extension **.php** !

## I. L'environnement

Alors attention, ce n'est pas parce que l'adresse de la page contient `.php` que c'est du PHP que votre navigateur manipule. Cela veut juste dire qu'on accède à un fichier qui, sur le serveur, *contient* du script PHP. Et ce script PHP va en fait être utilisé pour fournir à votre navigateur quelque chose qu'il comprend, soit... du **HTML**, principalement. 🍌  
Mais pour que PHP puisse fournir du **HTML**, il vous faut quelque chose qui comprenne le PHP... un **serveur** !

### I.1.2. Pour discuter avec un serveur : le protocole HTTP

Quand un client discute avec un serveur web, ils utilisent une "forme de discussion" spécifique : le protocole **HTTP**<sup>1</sup>. Pour simplifier, c'est juste une manière de formuler les demandes (appelées **requêtes**) et les **réponses**.

Continuons un instant avec notre exemple de restaurant.

Quand vous êtes dans un restaurant, soit vous savez ce que vous allez commander, soit vous regardez la carte et choisissez en conséquence. Vous passez donc votre commande.

Imaginons que vous avez commandé un steak de cheval avec des frites. Vous ne vous attendez pas à ce que l'on vous serve un steak cru et des pommes de terre non seulement crues aussi, mais en plus entières : vous attendez que la cuisine vous transforme les divers ingrédients de base en une viande cuite à point et des frites croustillantes, bref, un plat que vous pourrez manger.

C'est un peu similaire sur Internet quand vous souhaitez afficher une page.

Votre navigateur est ouvert, et soit vous savez où vous voulez aller et vous entrez l'adresse directement, soit vous regardez les liens proposés et en choisissez un. Quand vous validez l'adresse que vous avez tapée ou que vous cliquez sur un lien, vous envoyez une *requête*. Le serveur va se charger de la "cuisine", soit d'interpréter les scripts qu'il est en mesure d'interpréter, et il vous répondra avec une page que vous pourrez voir dans votre navigateur. Ainsi, vous ne recevez pas de code que votre navigateur ne comprendra pas.

Vous l'aurez compris :

- la cuisine du restaurant, c'est le serveur d'hébergement
- le client, c'est votre navigateur

Tout comme dans un restaurant vous faites votre commande au serveur (ne pas confondre la personne et la machine 🍌), avec votre navigateur vous envoyez la requête par **HTTP**. Voici donc une autre raison de la nécessité d'avoir un serveur sur votre machine si vous faites du PHP.

?

Mais quand je faisais simplement du HTML / CSS, je n'avais pas de serveur, donc je n'utilisais pas **HTTP**! Pourquoi ça fonctionnait?

Les navigateurs connaissent et utilisent **HTTP**, mais peuvent aussi accéder aux fichiers de votre ordinateur sans avoir besoin de ce protocole. Et comme le HTML est compris par le navigateur, vous pouvez afficher des pages ainsi.

Mais pour des scripts en PHP, vous devez les faire "passer en cuisine" si vous ne voulez pas de la

---

1. les self-services n'existent pas

## I. L'environnement

~~viande et des patates crues~~ que votre code soit considéré comme du texte brut. Essayez d'ouvrir un fichier PHP en utilisant le menu de votre navigateur ou le raccourci **Ctrl** + **O**. Soit vous verrez le code comme du texte, soit vous aurez une fenêtre de téléchargement vous demandant où enregistrer votre fichier. Et ce n'est pas vraiment ce que nous souhaitons, n'est-ce pas ?



Ceci implique que, au contraire des fichiers HTML, les fichiers PHP ne peuvent pas être ouverts pour être interprétés en faisant un double clic dessus. Vous **devez** passer par le protocole **HTTP** pour discuter avec le serveur.

### I.1.2.1. Une autre raison encore : le cas d'AJAX<sup>2</sup>

Peut-être avez-vous entendu parler d'**AJAX**, et vous vous préparez à en utiliser pour votre site. **Sachez qu'**AJAX** ne peut être utilisé qu'avec un serveur.**

Si votre navigateur peut accéder aux fichiers de votre ordinateur sans utiliser **HTTP** (en fait, sans envoyer de requête), ce n'est pas le cas de l'objet XMLHttpRequest fourni par votre navigateur, qui est à la base d'**AJAX**, et qui (comme son nom le laisse penser) construit des requêtes. On ne parle pas de requête pour accéder à un fichier sur votre machine alors que vous n'avez pas de serveur.

Ainsi, si vous vous préparez à utiliser **AJAX**, il vous faut aussi un serveur, même sans PHP.

## Conclusion

Maintenant que nous savons qu'il nous faut un serveur, il ne manque plus qu'à l'installer !

Rendez-vous au chapitre suivant. 🍊

---

1. HyperText Transfert Protocol  
2. Asynchronous Javascript and XML

## I.2. Avant d'installer...

### Introduction

Les diverses solutions ont toutes des besoins et des ennemis communs. Parmi ceux-ci, divers programmes ou parties de programmes qui accaparent le port 80, port utilisé pour la communication HTTP.

Ici, nous vous parlons des programmes qui posent le plus souvent problème, nous vous expliquons comment faire pour les minimiser sinon les résoudre, et nous donnons aussi quelques pistes pour les cas où le programme qui pose problème n'est pas parmi les plus courants mentionnés ici – hé, on ne peut de toute manière penser à tout... 🍊

i

Ce qui figure dans ce chapitre peut aussi vous aider en cas de syndrome de l'icône de WAMP qui reste orange. 🍊

### I.2.1. IIS

IIS est à Windows ce qu'Apache est à LINUX, soit un gestionnaire de serveur web. Mais si vous êtes en train de lire ceci, vous utilisez WAMP, vous n'aurez donc probablement pas besoin de ce programme, et pourrez le désinstaller sans autre.

IIS est le programme en cause si le test du port 80 vous parle de **microsoft - httpapi/2.0** (ou approchant) sous Windows Vista et 7, **Microsoft IIS/8.0** sous Windows 8 et **Microsoft IIS/10.0** sous Windows 10.

Pour désinstaller ce programme qui ne vous servira pas à grand-chose sinon vous ennuyer, rendez-vous dans la liste des *Programmes et fonctionnalités*. Sous Windows Vista ou 7, passez par le Panneau de Configuration, puis cliquez sur la rubrique correspondante.

Sous Windows 8 ou 10, vous pouvez accéder au panneau de configuration en effectuant un clic droit sur le bouton démarrer et en sélectionnant *Panneau de configuration* dans le menu contextuel.

Sinon, vous pouvez aussi utiliser le raccourci 

; + **R**, puis tapez `appwiz.cpl`.

Maintenant, dans la fenêtre qui s'ouvre et qui liste les programmes installés, il faut regarder la partie de gauche de la fenêtre, où vous avez un lien *Activer ou désactiver des fonctionnalités Windows*. Cliquez dessus.

Dans la nouvelle fenêtre qui s'affiche, cherchez et décochez *Instance principale Web des services*

## I. L'environnement

*Internet (IIS)* (Windows 7, 8) ou *Internet Information Services* (Windows 10). Validez avec OK. Il vous faudra probablement redémarrer votre ordinateur pour que les changements soient totalement effectifs.

Si vous souhaitez garder IIS installé, il faudra désactiver le démarrage automatique du service (voir plus bas).

### I.2.2. Skype

Pour faire en sorte que Skype ne gêne pas, il suffit de décocher la case *Utiliser les ports 80 et 443 comme alternative* dans les options de connexion. Vous n'avez pas besoin de vous connecter à un compte pour y accéder.

Si vous êtes déjà connecté, vous trouverez les options de connexion dans la partie *Avancées*.

Dans tous les cas, il vous faudra redémarrer Skype pour que ces changements soient pris en compte.

### I.2.3. SQL Server Reporting Services

Si vous le souhaitez, vous pouvez sans autre désinstaller ce composant, qui permet de se connecter au serveur SQL pour récupérer des informations sur l'état du serveur, des bases de données, etc..

Pour désinstaller SQL Server Reporting Services, il vous faudra modifier votre installation (*Panneau de configuration > Programmes et fonctionnalités*, sélectionner *SQL Server* dans la liste, puis cliquer sur *Modifier* ou, selon la version, *Désinstaller/Modifier* qui apparaît au sommet de la liste) de MS SQL Server.

### I.2.4. Vérifier si un autre programme utilise le port 80

Il n'est pas impossible qu'un autre programme soit déjà en écoute sur le port 80. Après avoir traqué les plus connus, nous allons voir comment pister les plus discrets. 🐼

Lancez l'invite de commande de Windows **en tant qu'administrateur**, et tapez `netstat -ano | find "0.0.0.0:80"`. Retenez le dernier nombre de la ligne.

Accédez au gestionnaire des tâches Windows, et rendez-vous dans l'onglet *Services*. Cherchez le nombre que vous avez depuis l'étape précédente dans la colonne *PID*. Quand vous l'avez trouvé, notez le nom du service, puis faites un clic droit sur la ligne correspondante, et choisissez *Arrêter le processus*.

Il arrive parfois que ce ne soit pas un service, mais un programme, et vous ne le trouverez donc pas dans l'onglet *Services*.

Rendez-vous plutôt dans *Processus*. Il faut tout d'abord afficher la colonne *PID*. Tout en restant sur l'onglet, allez dans le menu *Affichage > Sélectionner des colonnes...*, et cochez *PID* (premier dans la liste). Validez avec OK.

## I. L'environnement

Cherchez le processus avec le PID correspondant au nombre noté plus tôt, et sélectionnez la ligne en cliquant dessus. Notez éventuellement le nom du processus, cela pourra vous servir. Il ne vous reste plus qu'à choisir *Arrêter le processus* dans le menu contextuel.

### I.2.4.1. Prévenir vaut mieux que guérir

Le service ou le programme que vous venez d'arrêter peut être lancé au démarrage de Windows. Dans ce cas, il serait peut-être intéressant de désactiver ce lancement.

Dans le cas d'un programme, regardez dans ses options s'il n'y a pas quelque part une option qui dit quelque chose comme "Lancer au démarrage de Windows". Si c'est le cas et que cette option est activée, vous pouvez probablement la désactiver. Vous pourrez toujours lancer le programme en allant le chercher dans le menu Démarrer, et vous pourrez même réactiver l'option si vraiment.

S'il s'agissait d'un service, on passe à la suite !

### I.2.5. Désactiver les services plutôt que les désinstaller

Si vous ne voulez ou ne pouvez pas désinstaller les composants qui posent problème, vous pouvez aussi faire en sorte que les services ne démarrent qu'à votre demande. Il vous faudra cependant effectuer les manipulations en mode administrateur.

Commencez par ouvrir le panneau de configuration, puis cliquez sur *Outils d'administration* et choisissez *Services*.

Une fois de plus, vous aurez une nouvelle fenêtre qui s'ouvrira. Dans celle-ci, la liste des services disponibles s'affiche. Cherchez le service que vous souhaitez (par exemple celui dont le nom correspond à ce que vous avez noté à l'étape précédente 🍊), puis effectuez un clic droit dessus, et cliquez sur *Propriétés*. Dans l'onglet *Général*, vous trouverez une liste de choix *Type de démarrage*. Sélectionnez donc *Manuel*.

Certains services peuvent être démarrés par la console d'administration du "programme parent" (SQL Server reporting Services), d'autres démarrent en même temps que le programme parent (SQL Server Express avec Visual Studio), et d'autres encore vous forceront à repasser par le gestionnaire de services, comme nous venons de le faire. Seulement, cette fois, vous sélectionnez *Démarrer* dans le menu contextuel du service.

## Conclusion

Nous avons maintenant fait le maximum pour pouvoir installer et utiliser WAMP sans problème.

Si vous avez une pile AMP encore installée, la dernière chose à faire avant la nouvelle installation c'est de la désinstaller (en ayant fait les sauvegardes qui s'imposent, évidemment). 🍊

## I.3. WAMP

### Introduction

Parmi toutes les solutions "tout-en-un" de serveurs pour Windows, vous trouverez **WAMP**. Outre le fait que cette solution fonctionne dès la fin de son installation, elle vous permet d'avoir plusieurs versions de PHP, de MySQL et même d'Apache, afin de pouvoir être au plus près de votre environnement de déploiement.

Le paramétrage de PHP et d'Apache peut se faire directement au travers du menu accessible depuis l'icône de la barre de statut, ce qui vous évitera d'avoir à aller manipuler les fichiers de configuration vous-même dans les cas les plus courants.

*i*

On dit **WAMP**, mais ce nom est générique et désigne toutes les solutions qui installent Apache, PHP et MariaDB / MySQL sur un environnement Windows. Le vrai nom de ce que l'on vous propose d'installer est **WampServer**. Mais pour les mêmes raisons que l'on dit «frigo» au lieu de «frigoridaire», on va parler de « **WAMP**» pour désigner «WampServer». 🍊

Si vous mettez à jour **WAMP** ou que vous migrez d'une autre solution à **WAMP**, vous pouvez lire la partie suivante qui vous donnera des conseils utiles pour la sauvegarde des données.

Si vous installez **WAMP** pour la première fois et comme unique solution serveur, vous pouvez directement passer à l'installation.

!

Depuis la version 3.2.0, WampServer va forcément installer MariaDB, et proposer MySQL en option. ***Sachez que vous n'avez PAS absolument besoin de MySQL***, MariaDB étant une alternative *open source* à et compatible avec MySQL, qui est propriété d'Oracle. Plusieurs points à noter ici :

- à terme, MySQL ne sera plus proposé qu'en module séparé pour WampServer, sinon plus du tout ;
- installer MariaDB **et** MySQL et les conserver les deux vous force à être attentifs à quelle solution gère vos bases de données et comment vous y connecter (cf. [la FAQ PHP à ce propos](#) ) ;
- vu ce qui précède, ce tutoriel ne parlera pas de l'un ni de l'autre nommément, mais de **SGBD** pour désigner les deux ensemble comme l'un ou l'autre séparément.

### I.3.1. Sauvegarder, sauvegarder, sauvegarder !

Il y a plusieurs choses dont il faut tenir compte quand vous passez d'une solution à une autre ou d'une version à l'autre, c'est que toutes les données et tous les paramètres ne sont pas nécessairement conservés. Il vous faut donc bien penser à sauvegarder ce qui suit:

- vos fichiers de scripts [↗](#) ;
- les bases de données et leur contenu [↗](#) ;
- les configurations :
  - de PHP [↗](#) ;
  - du SGBD [↗](#) ;
  - d'Apache [↗](#) :
    - les alias [↗](#) ;
    - les hôtes virtuels [↗](#) .

#### I.3.1.1. Sauvegarder les scripts

Avec les serveurs, il y a un dossier où se trouvent la plupart de vos œuvres. Ce dossier a différents noms en fonction de ce que vous utilisez. Parmi les plus courants, il y a:

- `www`
- `htdocs`
- `httpdocs` avec sa version sécurisée `httpsdocs`
- `httproot` avec sa version sécurisée `httpsroot`

Faites une archive des dossiers qui vous concernent. Elles seront à extraire dans le dossier `www` de votre nouvelle installation de WAMP.

#### I.3.1.2. Sauvegarde des bases de données

Si vous utilisiez un ou plusieurs SGBD, il faut savoir que mettre à jour la version ne permet pas nécessairement de récupérer ses données. Il est donc très vivement conseillé de faire une sauvegarde du serveur entier.

WAMP, comme beaucoup de solutions, est livré avec une interface de gestion des bases de données : phpMyAdmin. Cette interface vous permet d'importer plusieurs tables en une fois et, évidemment, d'exporter aussi ces tables en une fois.

##### I.3.1.2.1. Les données "courantes"

Une fois que vous êtes sur la page d'accueil de phpMyAdmin (après vous être connecté au besoin), rendez-vous dans la partie *Exporter*. Pas besoin de faire trop dans la dentelle, vous pouvez laisser les paramètres par défaut. Toutes vos bases de données seront exportées dans un seul et unique fichier SQL que vous pourrez réimporter une fois WAMP installé.



### I.3.1.2.2. Les utilisateurs

C'est un point à ne pas oublier, surtout si vous aviez un utilisateur par site avec l'ancienne solution! 🍊

Toujours depuis la page d'accueil de phpMyAdmin, cliquez sur l'onglet *Utilisateurs*. Cochez-les tous sauf les deux *N'importe quel* et les deux ou trois *root*, qui seront recréés automatiquement. Cliquez ensuite sur *Exporter*. L'export des utilisateurs ne se fait pas dans un fichier, il vous faut donc copier-coller ce qui s'affiche dans un éditeur de texte.

### I.3.1.3. Sauvegarder la configuration

#### I.3.1.3.1. Configuration de PHP



La configuration de PHP ne peut être utilisée que pour cette version et ses révisions. Si vous avez par exemple un `php.ini` pour PHP 5.3.6, il sera utilisable pour toutes les versions 5.3 (moyennant remplacement des chemins qui y sont mentionnés), mais pas pour les versions 5.4 ou plus, ni pour 5.2 ou moins.



Mais alors à quoi sert de conserver ces paramètres?

Nous pouvons utiliser plusieurs versions de PHP avec WAMP, donc vous pourrez installer une version pour laquelle ces paramètres seront utiles.

Si vous comptez laisser tomber cette version de PHP, conservez tout de même ces fichiers de configuration: vous pourrez les comparer avec ceux fournis et ainsi reporter les différences.

La configuration de PHP se limite souvent au fichier `php.ini`, mais il n'est pas impossible qu'il y en ait plusieurs. Vérifiez donc lesquels sont utilisés en accédant au `phpinfo` de votre serveur actuel. Si vous utilisez aussi PHP en ligne de commandes, lancez une fois la commande `php --ini`.

Dans le cas où vous avez deux `php.ini` différents en ligne de commandes et pour Apache, mémorisez bien lequel est pour quel environnement.

Dans WAMP, le fichier pour Apache s'appelle `phpForApache.ini`, tandis que le fichier pour la ligne de commandes est nommé `php.ini`. Les deux sont à placer au même endroit que `php.exe`, ce qui fait, pour WAMP, `%WAMPDIR%\bin\php\php%PHP_VERSION%`<sup>1, 2</sup>.

#### I.3.1.3.2. Configuration du SGBD

Pour le **SGBD**, c'est simple: il y a un fichier `my.ini` qui se trouve dans `%WAMPDIR%\bin\%SGBD_NAME%\%SGBD_NAME%\%SGBD_VERSION%`<sup>3, 4, 5</sup>. Il n'est pas toujours réutilisable tel quel, mais pour des changements de versions mineures peu éloignées, il pourrait bien n'y avoir que des chemins à adapter.

### 1.3.1.3.3. Configuration d'Apache

Les installations d'Apache possèdent un dossier `conf` qui contient la majeure partie de la configuration. La configuration d'Apache pose un peu moins de problèmes entre les versions, néanmoins il n'est pas conseillé d'écraser la configuration "de base" de WAMP par une ancienne. Si vous savez que vous avez des particularités, vous saurez probablement les reporter dans votre nouvelle installation. 🍌

**1.3.1.3.3.1. Les alias** WAMP possède un dossier `%WAMPDIR%\alias`<sup>6</sup> au même endroit que le dossier `www`, qui contiendra autant de fichiers que d'alias définis. Un de ceux qui s'y trouve contient de quoi accéder à phpMyAdmin:

```
1 Alias /phpmyadmin "C:/wamp/apps/phpmyadmin4.1.14/"
2
3 # to give access to phpmyadmin from outside
4 # replace the lines
5 #
6 # Require local
7 #
8 # by
9 #
10 # Require all granted
11 #
12
13 <Directory "C:/wamp/apps/phpmyadmin4.1.14/">
14     Options Indexes FollowSymLinks MultiViews
15     AllowOverride all
16     <IfDefine APACHE24>
17         Require local
18     </IfDefine>
19     <IfDefine !APACHE24>
20         Order Deny,Allow
21         Deny from all
22         Allow from localhost ::1 127.0.0.1
23     </IfDefine>
24     php_admin_value upload_max_filesize 128M
25     php_admin_value post_max_size 128M
26     php_admin_value max_execution_time 360
27     php_admin_value max_input_time 360
28 </Directory>
```

Listing 1 – Fichier `phpmyadmin.conf` fourni avec WAMP 2.5, installé dans `C:\wamp`  
Il vous faut mettre chaque ligne ressemblant à la toute première dans son propre fichier. Ensuite, s'il y a une ligne `<Directory ...>` comme la ligne 13 avec le même chemin que dans la première ligne (ici, vous avez les deux fois `C:/wamp/apps/phpmyadmin4.1.14/`), tout ce qu'il y a entre `<Directory ...>` et `</Directory>` (y compris) doit être mis à la suite dans le fichier que vous avez créé. Prenons un exemple avec une structure un peu plus complexe :

```
1 Alias /phpmyadmin "C:/wamp/apps/phpmyadmin4.1.14/"
2 Alias /webgind "C:/wamp/apps/webgind7.1.0/"
3
4 <Directory "C:/wamp/apps/phpmyadmin4.1.14/">
5     # ...
6 </Directory>
7 <Directory "C:/wamp/apps/webgind7.1.0/">
8     # ...
9 </Directory>
```

Vous aurez deux fichiers contenant :

- la ligne 1 et tout ce qu'il y a entre les lignes 4 à 6 comprises ;
- la ligne 2 et tout ce qu'il y a entre les lignes 7 à 9 comprises.

Notez que les alias fournis avec votre pile AMP actuelle et qui seront fournis avec votre nouvelle solution (ceux pour phpMyAdmin, WebGrind ou encore SQL Buddy par exemple) n'ont pas besoin d'être sauvegardés: la nouvelle installation va les recréer adaptés.

#### 1.3.1.3.4. Les hôtes virtuels (<VirtualHost>)

Si vous en aviez créé auparavant, il serait intéressant de les retrouver plus tard. 🍊

Le plus couramment, les hôtes virtuels sont créés dans un fichier %APACHEDIR%\conf\extra\httpd-vhosts.conf<sup>7</sup>, et se présentent sous la forme de blocs <VirtualHost> comme dans l'exemple suivant:

```
1 <VirtualHost *:80>
2     ServerName zds.local
3     ServerAlias www.zds.local dev.zds.local
4     DocumentRoot "C:\wamp\www\zds"
5     <Directory "C:\wamp\www\zds">
6         AllowOverride All
7     </Directory>
8 </VirtualHost>
```

Listing 2 – Un exemple de définition d'hôte virtuel

Vous pouvez conserver l'entier du fichier httpd-vhosts.conf, il ne contient pas souvent d'autres configurations qui poseraient problème plus tard.

---

1. %WAMPDIR% est à remplacer par votre chemin d'installation de WAMP  
2. %PHP\_VERSION% est à remplacer par la version de PHP concernée  
3. %WAMPDIR% est à remplacer par votre chemin d'installation de WAMP  
4. %SGBD\_NAME% est à remplacer par le nom du SGBD en minuscules  
5. %SGBD\_VERSION% est à remplacer par la version de votre SGBD utilisée  
6. %WAMPDIR% est à remplacer par votre chemin d'installation de WAMP  
7. %APACHEDIR% est à remplacer par le chemin de l'installation d'Apache pour laquelle vous sauvegardez la configuration

## 1.3.2. Mise à jour de WAMP

Si vous aviez une ancienne version de WAMP et que vous souhaitez la mettre à jour, nous allons voir comment faire ici.

i

Si vous mettez à jour WAMP uniquement pour avoir une version plus récente de PHP ou de votre SGBD, sachez que vous n'en avez pas nécessairement besoin: regardez la dernière partie du tutoriel [↗](#) qui vous apprendra à utiliser plusieurs versions de PHP avec votre installation actuelle. 🎉

WAMP existe maintenant depuis un bon moment, et les versions se sont succédées, souvent au gré des versions majeures de PHP ou d'Apache. Vous pourriez vouloir vous mettre à jour comme vous le faites avec vos autres logiciels, ce qui est possible. Cependant, ce n'est pas un processus qui se fait en un clic ! Il y a des précautions à prendre et, selon votre version de WAMP, la marche à suivre n'est pas la même que pour d'autres.

On parle volontiers de mises à jour **majeures** et **mineures** quand on est dans le domaine de l'informatique. Pour WAMP, les mises à jour *majeures* se traduisent par un changement dans le premier chiffre de la version, et c'est souvent synonyme de fonctionnalités supplémentaires, de support élargi ou de compatibilité avec des outils plus récents. A contrario, les mises à jour *mineures* ajoutent des fonctionnalités simples. Il y a encore ce qu'on appelle les **patches** (*patch* au singulier), qui sont là pour corriger des bugs.

Il fut un temps où les patches avaient deux systèmes de numérotation, celui expliqué ci-dessus, et un autre qui utilisait des minuscules après le dernier nombre. Mis en place pour les versions 2.0 à 2.2 de WAMP, il n'est actuellement plus utilisé.

### 1.3.2.1. Mises à jour majeures

#### 1.3.2.1.1. Mettre à jour depuis WAMP 2.5 ou plus vieux

Dans ce cas, malheureusement, il n'y a pas vraiment d'autre possibilité que de faire les sauvegardes, puis de désinstaller complètement la version que vous avez actuellement, et ensuite d'installer la nouvelle version, puis d'y importer vos scripts et données. C'est fastidieux, mais c'est la manière la plus sûre.

Prenez le temps de bien lire les sections précédente et suivante pour ne rien oublier.

#### 1.3.2.1.2. Mettre à jour depuis WAMP 3.0 vers WAMP 3.2

Là, c'est un peu plus facile : il a été prévu des packs de mise à jour ! 🍊

Cependant, il faut bien faire attention, parce qu'il y a des versions **x86** et **x64**, et évidemment, il vous faut donc savoir **quelle version exacte vous souhaitez mettre à jour**. Mélanger les versions ne va pas vraiment être concluant, même si au premier abord ça pourrait marcher.





Pour vérifier quelle version vous avez, effectuez un clic droit sur l'icône de WAMP et sélectionnez "A propos". Une petite fenêtre s'affichera, vous renseignant sur la version de WAMP ainsi que sur l'architecture pour laquelle il est prévu—sachant que 64 bits correspond à **x64** et 32 bits à **x86**. Retenez bien l'information. 🍊

Les mises à jour ont les mêmes contraintes que pour l'installation de WAMP 3.0, soit notamment que toutes les versions de Microsoft Visual C++ nécessaires doivent être installées avant de faire la mise à jour, et le tout lancé avec le clic droit > *Exécuter en tant qu'administrateur*. Voici déjà les liens vers les installeurs pour les versions suivantes de Microsoft Visual C++


Redistribuable.




C'est l'architecture de votre version de Windows qui dicte s'il vous faut, *en plus des versions x86*, la version x64. Donc si vous avez un Windows x64, il vous faut 2 versions de chaque paquetage ci-dessous, soient 12 installations au total !

1. [Microsoft Visual C++ 2022 x86](#)  ,  
[Microsoft Visual C++ 2022 x64](#)  .  
**À considérer comme la dernière mise à jour de Microsoft Visual C++ 2015 ou 2017.**  
Choisir dans la page pour x86 ou x64—contient les fichiers **vcruntime140.dll** et **msvcr140.dll** ;
2. [Microsoft Visual C++ 2013 x86](#)  et [Microsoft Visual C++ 2013 x64](#)  et leur 5<sup>e</sup> pack de mise à jour  .  
Contient le fichier **msvcr120.dll** ;
3. [Microsoft Visual C++ 2012 update 4](#)  .  
Choisir dans la page pour x86 ou x64—contient le fichier **msvcr110.dll** ;
4. [Microsoft Visual C++ 2010 SP1 x86 et x64](#)  .  
Choisir dans la page pour x8 ou x64 – contient le fichier **msvcr100.dll** ;
5. [Microsoft Visual C++ 2008 SP1 x86](#)  ,  
[Microsoft Visual C++ 2008 SP1 x64](#)  .  
Contiennent le fichier **msvcr90.dll** ;
6. Au besoin, [la mise à jour du runtime C universel](#)  pour votre version exacte de Windows—contient le fichier **api-ms-win-crt-runtime-l1-1-0.dll**.



Microsoft propose sur une unique page la plupart des liens de téléchargements  , et ceux-ci sont parfois plus à jour que sur cette page du tutoriel. Pensez à vous y référer en cas de problème avec les liens ci-dessus.

N'oubliez pas d'installer avec le clic droit > *Exécuter en tant qu'administrateur*. Une fois les installations effectuées et l'ordinateur éventuellement redémarré, vous pourrez prendre la mise à jour, **elle aussi à installer avec le clic droit > Exécuter en tant qu'administrateur**. Trêve de blabla, voici de quoi aller chercher [l'installateur de mise à jour](#)  —choisissez bien la version que vous téléchargez. 🍏

### 1.3.2.2. Mises à jour mineures

Pour WAMP, c'est arrivé souvent pour sa version 2, où il y a eu 2.0, puis 2.1, 2.2, etc. jusqu'à 2.5. Cependant, la procédure de mise à jour était la même que lors de changements de version majeure.

### 1.3.2.3. Patches

i

Le mainteneur de WAMP préconise de toujours appliquer les patches.

Comme pour tout programme, il arrive qu'il y ait des possibilités qui ne sont pas totalement fonctionnelles. Heureusement, il y a aussi des mises à jour qui corrigent ces bugs, même s'il faut aller se renseigner. Au niveau du numéro de version, c'est le troisième chiffre qui change.

Il peut arriver que ce ne soit pas qu'un exécutable à lancer, mais qu'il y ait des modifications à faire dans les fichiers de configuration. C'était notamment le cas pour [l'évolution de WAMP 2.5](#) [↗](#), où il y a eu pas moins de 18 versions ! Pour WAMP 3, au moment où ceci est écrit, les modifications sont pour l'heure entièrement automatisées.

Autre point intéressant : ces paquetages de mise à jour sont efficaces pour les versions **x86** comme **x64**, ce qui veut dire que le téléchargement sera le même quelle que soit votre version de WAMP. Souvent, ce fait se traduit par une version "particulière" : **x86\_64**. Si vous voyez donc des téléchargements pour WAMP signalés avec cette version, cela signifie qu'ils fonctionneront sans souci avec votre WAMP—pour autant que vous ayez pris un add-on qui soit pour la même version mineure. 🍏

Vous trouverez le dernier patch pour la dernière version de WAMP sur [cette page](#) [↗](#). Je vous propose de vous y rendre régulièrement si vous souhaitez avoir les derniers correctifs.

### 1.3.3. Installation

Pour commencer, nous allons avoir besoin de pas moins de **six** installeurs, et même **onze** dans le cas d'une version 64 bits de Windows.

!

Il ne faut pas prendre les versions **x64** parce que vous prenez WampServer 64 bits : il faut prendre les versions **x64** parce que *vous avez une version 64 bits de Windows*. Donc **même si vous prenez WampServer 32 bits avec Windows 64 bits, il vous faut les versions x64**.

Aussi, pour Windows **x64**, **il faut quand même installer les paquetages x86**.

1. Les installeurs pour les versions suivantes de Microsoft Visual C++ Redistribuable :

i

Vu que c'est un des soucis courants que d'oublier un des paquetages ou encore que certains sont déjà installés, il y a un utilitaire qui va vous dire ce qui n'est pas installé et qui le devrait [↗](#).

Si vous faites une installation sur un ordinateur "propre", il existe une archive zip pour **x86** [↗](#) et une archive zip pour **x64** [↗](#).

- a) Microsoft Visual C++ 2022 **x86** [↗](#),  
Microsoft Visual C++ 2022 **x86** [↗](#) ;
- b) Microsoft Visual C++ 2013 **x86** [↗](#),  
Microsoft Visual C++ 2013 **x64** [↗](#)  
et leur 5<sup>e</sup> pack de mise à jour [↗](#) ;
- c) Microsoft Visual C++ 2012 update 4 [↗](#) – choisir dans la page pour **x86** ou **x64** ;

## I. L'environnement

- d) [Microsoft Visual C++ 2010 SP1 x86 et x64](#) – choisir dans la page pour x8 ou x64 ;
- e) [Microsoft Visual C++ 2008 SP1 x86](#) ,  
[Microsoft Visual C++ 2008 SP1 x64](#) ;
- f) Au besoin, [la mise à jour du runtime C universel](#) pour votre version exacte de Windows.

**i**

Comme signalé précédemment, Microsoft propose [une page avec la plupart des liens de téléchargement](#) conservés à jour. Pensez à vous y référer en cas de problème avec les liens ci-dessus.

- 2. L'installateur de la version complète de [WampServer 3.2.8](#) ou la version la plus récente disponible sur cette même page.

**!**

### Compatibilité

À partir de la version **3.1**, l'installateur de WAMP n'est **plus compatible avec Windows Vista et moins récent**.

Si vous souhaitez utiliser WAMP 3 avec un de ces systèmes d'exploitation, rendez-vous sur [le dépôt SourceForge de WampServer](#) pour télécharger une ancienne version qui soit compatible avec votre OS (mais pas plus vieille que la 2.5), puis suivez les instructions de [la section «Mise à jour de WAMP» ci-avant](#) , et éventuellement regardez [comment ajouter des versions de PHP](#)

Maintenant, commencez par installer toutes les versions de Visual C++ en faisant clic droit > *Exécuter en tant qu'administrateur* sur les programmes d'installation que vous avez téléchargés. Cela permet de rendre l'installation disponible pour tous les utilisateurs de l'ordinateur. S'il vous est dit que le paquetage est déjà installé, choisissez de réparer l'installation. Il est possible qu'il vous faille redémarrer votre ordinateur : il n'est pas nécessaire de le faire entre chaque installation, mais faites-le une fois pour toutes avant d'aller plus loin.

La prochaine étape est l'installation de WAMP elle-même. Là aussi, il faut la lancer avec clic droit > *Exécuter en tant qu'administrateur*. Installer WAMP en tant qu'administrateur permet d'éviter des problèmes de droits quand vous le lancez, d'installer pratiquement n'importe où et prépare aussi les services Apache et **SGBD** correctement.

Nous disions que lancer l'installation de WAMP en tant qu'administrateur permettait de l'installer n'importe où sur votre machine, mais il faut absolument éviter de le mettre dans les dossiers `C:\Program Files\wamp` ou `C:\Program Files (x86)\wamp`. Ces deux dossiers ont des droits particuliers qui vont poser problème quand vous travaillerez sur vos scripts et que vous voudrez y accéder pour voir le résultat. En conséquence, préférez toujours une installation à la racine d'un lecteur, comme `C:\wamp`, par exemple.

À un moment de l'installation, il vous est demandé de spécifier quel navigateur vous utilisez par défaut. Comme nous sommes sous Windows, WAMP va prendre le seul navigateur dont il est certain de la présence : Internet Explorer. Cependant, si vous souhaitez utiliser un autre navigateur, il faut savoir où il se trouve. Voici de quoi choisir rapidement :

Che- Nain vide gal'exé- teon- table	C:\Pro gram Files (x86)\Mo zilla Fi re fox\fi re fox.exe C:\Pro gram Files\Mo zilla Fi re fox\fi re fox.exe
Ch pli ca tion\chrome.exe <sup>1</sup>	C:\Users\%USER%\App Data\Lo cal\Google\Chrome\Ap pli ca tion\chrome.exe <sup>1</sup>
In- ter- net Ex- plo- rer	C:\Pro gram Files (x86)\In ter net Ex plo rer\iex plore.exe



	C:\Program Files\Internet Explorer\iexplore.exe
	C:\Program Files\Opera\Opera.exe
01	C:\Program Files\Opera\Opera.exe

TABLE I.3.2. – Principaux navigateurs sous Windows et leurs emplacements

Après le choix du navigateur, il vous sera aussi demandé quel éditeur de code vous souhaitez utiliser à la place du bloc-notes de Windows. Pour cette étape aussi, naviguez jusqu'à votre programme favori.



Tous les programmes ne fonctionneront pas nécessairement ainsi, à vous de tester. Vous pourrez toujours changer en vous rendant dans le fichier `%WAMPDIR%\wampmanager.conf` pour y remplacer le chemin du programme ne fonctionnant pas par un autre.



Et si je souhaite utiliser Edge ?

Là, c'est un peu plus compliqué, et ne fonctionne que sous Windows 10. Edge n'est pas un programme du même type que les navigateurs du tableau ci-dessus, et ne peut pas être choisi en naviguant vers son emplacement. Laissez Internet Explorer par défaut et terminez l'installation. Ensuite de quoi, avant de démarrer WAMP, il faudra aller modifier le fichier `%WAMPDIR%\wampmanager.conf` et remplacer le chemin vers Internet Explorer par le simple mot `Edge`.

Il faut aussi faire accepter à Edge le fait que vous allez utiliser des hôtes locaux. Pour ce faire, rendez-vous sur la page [about:flags](#) et cochez la seconde case à cocher intitulée "Autoriser le bouclage localhost".

### 1.3.3.1. Est-ce que les services Apache et SGBD sont bien installés ?

Si vous n'avez pas lancé WAMP après installation, c'est le moment où jamais !

Normalement (soit si vous n'avez pas changé les paramètres de l'**UAC**), Windows devrait vous demander de confirmer le lancement.

L'icône de WAMP va s'afficher dans la barre de statut, d'abord rouge, puis éventuellement orange (elle prend de toute façon cette couleur, mais c'est parfois trop bref pour qu'on le voie), puis (et surtout) verte. Pendant ce temps, l'invite de commandes va s'afficher brièvement quelques fois.

i

Si vous ne voyez pas l'icône de WAMP dans la barre de statut, pas de panique ! Certaines icônes ne sont pas toujours visibles, mais restent accessibles par un clic sur la flèche pointant vers le haut, un peu comme si c'était la dernière icône. Avec un cliquer-glisser, vous pouvez positionner l'icône de WAMP pour qu'elle reste accessible.

Si l'icône reste sur l'orange, il y a une vérification à effectuer, mais d'abord, on va faire en sorte que les menus soient en français.

Quelle que soit la couleur de l'icône, faites un clic droit dessus, puis pointez *Language* et sélectionnez *french*. Pas plus compliqué que ça 🍊

Apache et les **SGBD** fonctionnent comme des services sous Windows. Pour que ces services démarrent correctement et soient utilisables, il faut qu'ils soient installés correctement. Pour vérifier cela, cliquez sur l'icône de WAMP dans la barre de statut, puis pointez sur *Apache > Service*. Si les options *Arrêter le service* et *Redémarrer le service* sont grisées, cliquez sur *Installer le service*, juste en dessous.

De manière similaire pour le **SGBD**: cliquez sur l'icône de WAMP dans la barre de statut, puis pointez sur le nom du **SGBD > Service**. Si les options *Arrêter le service* et *Redémarrer le service* sont grisées, cliquez sur *Installer le service*, juste en dessous.

Vous pouvez désormais entrer l'adresse `http://localhost/` dans votre navigateur et contempler la page d'accueil de WAMP. Vous allez pouvoir commencer à coder en PHP 🍊

## Conclusion

Voilà, nous avons fait notre possible pour installer un environnement qui devrait être utilisable...

×

Hé, ho, mon icône WAMP reste orange ! Non, on n'a pas un environnement utilisable !

Dans le cas d'une icône de WAMP qui reste bloquée sur l'orange, voyez le chapitre «[Avant d'installer... ↗](#)» ci-avant.

Il ne fallait pas sauter les chapitres ni installer sans clic droit > Exécuter en tant qu'administrateur... 🍊

---

1. %USER% est à remplacer par le nom d'utilisateur Windows

# Conclusion

Maintenant qu'on a installé nos outils et que nous nous sommes assurés qu'ils fonctionnent, on va regarder comment les utiliser, afin de ne pas ~~vous blesser~~ bloquer votre machine. 🍊

# **Deuxième partie**

## **Utiliser son serveur**

# Introduction

Maintenant que nous avons installé l'environnement, nous allons voir comment l'utiliser, parce que, comme pour tout, il n'est pas vraiment possible de faire n'importe quoi et ça fonctionne



## II.1. Dans la vie de tous les jours...

### Introduction

Maintenant que nous avons installé l'environnement, nous allons voir comment l'utiliser, parce que, comme pour tout, il n'est pas vraiment possible de faire n'importe quoi et que ça fonctionne !



#### II.1.1. Paramétrer les composants

Maintenant que nous avons installé notre serveur, il va probablement falloir effectuer quelques réglages, comme activer des modules Apache ou des extensions PHP, mais aussi spécifier un paramètre important depuis PHP 5.3. WAMP permet de faire cela très simplement.

##### II.1.1.1. Paramétrer Apache

Apache est lui-même composé de **modules**, dont certains sont activés dès l'installation de WAMP, soit parce qu'ils sont souvent utilisés et que l'entreprise qui fournit WAMP a prédéfini qu'il serait activé, soit... parce qu'ils sont au cœur même de notre serveur. 🍊

En effet, l'un des modules permet l'utilisation de PHP avec Apache. Il est donc activé dès l'installation.

Mais ce n'est pas nécessairement le cas de tous les modules, notamment l'un d'entre eux qui revêt beaucoup d'importance si vous utilisez un framework comme CodeIgniter, Laravel ou Symfony, voire même plus simplement un CMS comme Joomla!, Drupal ou WordPress : celui qui permet de faire des réécritures d'URLs, afin de pouvoir éviter des adresses comme `?module=Item&action=show&idItem=23` et les voir sous la forme `/Item/show/23`, par exemple. J'ai nommé `mod_rewrite`, parfois désigné par son identificateur `rewrite_module`.

*i*

Si ces deux dénominations sont utilisées, c'est parce que `mod_rewrite` est le nom de fichier qui contient le module, et `rewrite_module` est le nom utilisé en interne pour désigner le module<sup>1</sup>.

Nous allons vérifier de suite s'il est activé afin de changer cela au besoin.

Une fois n'est pas coutume, rendez-vous dans le menu de WAMP accessible par un clic sur l'icône dans la barre de statut, puis pointez *Apache > Modules Apache*.

La liste qui apparaît est celle des modules existants pour Apache, mais les noms affichés sont les identificateurs. À côté de leurs noms, on peut voir trois choses :

- **Rien**  
C'est le cas le plus courant. Le module est disponible, mais pas activé.  
Dans le fichier de configuration, cela se traduit par une ligne commentée.
- **Une coche verte ( )**  
Second cas le plus courant, qui signifie que le module est activé, donc disponible.

## II. Utiliser son serveur

Dans le fichier de configuration, la ligne est décommentée.

- **Un i dans un cercle bleu** ( )

Cas particulier qui indique que le module ne peut pas être désactivé. C'est notamment utilisé pour les modules permettant d'exécuter une certaine version de PHP avec Apache (soient `php5_module`, `php7_module` ou `php_module`). Les modules dans cet état sont, avec les versions récentes de WAMP, regroupés en fin de liste.

- **Un carré rouge** ( )

Les fichiers nécessaires au fonctionnement du module ne sont pas présents. C'est donc à vous de les trouver et de les placer dans le dossier `%WAMPDIR%\bin\php\php%PHP_VERSION%\ext` si vous en avez besoin. Ceci fait, la ligne s'affichera comme dans le second cas.

- **Un panneau "attention"** ( )

Les fichiers nécessaires au fonctionnement du module sont présents, mais pas la ligne à décommenter dans le fichier de configuration. Il vous faudra aller l'insérer à la main si vous avez besoin de ce module.

Vérifiez l'état de `rewrite_module` dans la liste. Un clic dessus permet de passer d'un état à un autre. Notez que cela va faire redémarrer Apache automatiquement, et Apache uniquement, ce qui veut dire que l'icône ne passera que par l'orange. Si l'icône redevient verte, vous pouvez immédiatement utiliser les fonctionnalités qu'offre le module.

### II.1.1.2. Paramétrer PHP

WAMP permet de paramétrer deux choses distinctement par son menu. Si les possibilités et le fonctionnement sont similaires à celui des modules Apache, ce n'est pas pour rien qu'il y a deux menus séparés. Par contre, les possibilités sont un peu différentes de ce que l'on peut observer pour Apache. Qui plus est, selon le menu, certaines options n'y sont pas observables.

- **Rien**

De même que pour Apache, l'extension est disponible, mais pas activée.

Dans le fichier de configuration, cela se traduit par une ligne commentée.

- **Une coche verte** ( )

Là aussi, cela veut dire que l'extension est activée.

Dans le fichier de configuration, la ligne est décommentée.

- **Un triangle vert pointant vers la droite** ( ) C'est l'indication que le paramètre ne supporte que certaines valeurs. Celles-ci sont généralement listées et fournies dans un nouveau sous-menu.

- **Un i dans un cercle bleu** ( )

Ici, ce n'est pour PHP qu'une indication sur une valeur particulière de la configuration.

- **Un panneau "attention"** ( )

Les fichiers nécessaires au fonctionnement du module ne sont pas présents. C'est donc à vous de les trouver et de les placer dans le dossier `%WAMPDIR%\bin\php\php%PHP_VERSION%\ext` si vous en avez besoin. Ceci fait, la ligne s'affichera comme dans le premier ou deuxième cas.

- **Un carré rouge** ( )

Les fichiers nécessaires au fonctionnement du module sont présents, mais pas la ligne à décommenter dans le fichier de configuration. Il vous faudra aller l'insérer à la main si vous avez besoin de ce module.

Voilà, maintenant regardons les menus disponibles qui nous intéressent.

## II. Utiliser son serveur

### II.1.1.2.1. Les extensions

PHP propose aussi des fonctions optionnelles, qui sont elles aussi activées ou non à l'installation. Si on parlait de modules avec Apache, on parle plutôt d'**extensions** avec PHP. L'une d'entre elles qu'il est intéressant d'activer est `php_openssl`, qui permet d'établir des connexions distantes en https avec `file_get_contents()` ou cURL, entre autres.

Affichez la liste des extensions PHP disponible en passant par l'icône de WAMP > PHP > *Extensions PHP*. La liste est bien plus courte que celles des modules Apache, vous n'aurez pas de mal à voir où se trouve `php_openssl` et l'activer au besoin. 🍊

Là aussi, WAMP va redémarrer Apache uniquement, donc l'icône passera à l'orange avant de redevenir verte.

### II.1.1.2.2. La configuration PHP

Outre le menu *Extensions PHP*, il y a aussi *Configuration PHP*. Les éléments de ce menu n'activent pas des fonctionnalités supplémentaires, mais plutôt changent le comportement de PHP en général. On y trouve des options relatives à la sécurité comme `allow_url_fopen`, qui détermine si on peut ouvrir un fichier hébergé sur un autre site avec les fonctions de lecture de fichiers (`fopen()`, `file_get_contents()` ou encore `readfile()` pour n'en citer que quelques unes), ainsi que `allow_url_include`, qui gère la possibilité d'inclure (avec `include`, `require`, `include_once` et `require_once`) des scripts sur des sites distants. C'est aussi là qu'on trouve `short_open_tags`, qui, si c'est activé, permet d'utiliser `<?>` au lieu de `<?php>`.

Il est justement conseillé de désactiver cette option depuis PHP 5.4, où elle était souvent activée parce que cela gérait aussi l'utilisation de `<?=>`. Or, toujours depuis PHP 5.4, `<?=>` est désormais utilisable que l'on ait activé `short_open_tags` ou non.

Ce dont il faut se rendre compte, c'est que la version courte `<?>` est aussi l'ouverture de la déclaration XML, et cela pose problème quand on génère un fichier XML avec PHP avec cette option activée. De fait, de nombreux CMS et *frameworks* actuels recommandent, voire exigent, que cette directive soit désactivée.

Certains paramètres n'acceptent pas n'importe-quelle valeur, mais une parmi un ensemble précis : c'est notamment le cas du paramètre `date.timezone`. Depuis la version 3.0.6, ce paramètre peut aussi être modifié depuis le menu *Configuration PHP*, et vous pouvez choisir le fuseau horaire de référence par continent, puis par ville. Pour les versions précédentes, il fallait éditer le fichier de configuration à la main.

Encore une fois, Apache devra redémarrer, donc icône à l'orange, puis verte.



Pourquoi est-ce que changer quelque chose pour PHP demande de redémarrer Apache ?

Parce que c'est Apache qui utilise PHP, et donc Apache qui doit savoir comment PHP est configuré. La configuration de PHP est lue au démarrage d'Apache, donc il faut le redémarrer pour que la nouvelle configuration soit prise en compte. 🍊

### II.1.1.3. Paramétrer votre SGBD

De manière similaire, vous pouvez accéder à certains paramètres de votre **SGBD** en passant par le menu de WAMP. Vous pouvez modifier la taille des buffers de tri, le temps de blocage sur les tables InnoDB, ou encore le détail des fichiers de logs. Une fois la modification enregistrée, c'est le service du **SGBD** qui va redémarrer, et donc faire varier la couleur de l'icône de WAMP.

Comme pour Apache qui utilise (entre autres) `httpd.conf` et PHP `php.ini`, vous pouvez



## II. Utiliser son serveur

accéder directement aux fichiers de configuration en mode texte depuis le menu de chaque partie de WAMP. Prenez cependant la précaution de vous renseigner sur quoi et comment changer la configuration de cette manière afin de ne pas "bloquer" votre serveur. 🍊

Enfin, si vous modifiez à la main les configurations d'Apache ou de PHP, n'oubliez pas de redémarrer Apache au minimum en passant par l'icône de WAMP > Apache > Service > Redémarrer le service.

### II.1.2. Organiser vos projets

Un serveur est, pour continuer avec l'image du restaurant, une cuisine. Pour cuisiner, il faut des ingrédients, et ces ingrédients, il faut les stocker dans un garde-manger.

Avec un serveur, on va dire que les ingrédients sont vos pages HTML, vos scripts PHP, vos fichiers CSS et JavaScript.

Il faut néanmoins savoir où aller les chercher. Pour WAMP, on ne va évidemment pas parler de garde-manger, mais de **racine serveur**. C'est le dossier où sont rangés les scripts qui seront disponibles pour vos sites. Ce dossier, par défaut, est le dossier `%WAMPDIR%\www`<sup>1</sup>. Pour ne pas trop charger, je ne parlerai que du «dossier `www`» par la suite. 🍊

#### II.1.2.1. Le dossier `www`

Dans le dossier `www`, vous pouvez mettre tout ce dont vous avez besoin pour construire vos sites. Vous pourrez y accéder en utilisant l'adresse `http://localhost/monFichier.php` ou `http://localhost/monAutreFichier.html`.

Seulement, vous l'imaginez bien, cela fonctionne tant que vous n'avez pas beaucoup de sites à gérer. En effet, un site contient plusieurs pages, et ces pages demandent plusieurs fichiers (des images, les fichiers CSS, des scripts JavaScript, etc.). Si vous voulez créer un autre site, imaginez le nombre de fichiers que ça fera...

De plus, certains fichiers seront nommés de la même manière d'un site à l'autre, mais sans avoir le même contenu, et ça, ce n'est ni possible ni pratique. En toute logique, on va donc mettre tous les fichiers pour chaque site dans des dossiers séparés. Ce qui fait que, pour accéder à un fichier de l'un ou l'autre site, l'adresse deviendra `http://localhost/monPremierSite/monFichier.php` ou `http://localhost/monSecondSite/monAutreFichier.html`.

Outre le fait que les fichiers sont désormais organisés, vous aurez remarqué que si vous accédez à `http://localhost/`, vous verrez les dossiers listés dans la partie *Vos projets*. Vous pouvez donc accéder à vos sites en passant par `http://localhost/`, puis en cliquant sur l'un des noms de dossier.

?

Mais ce ne sont pas des liens ! mais ça ne sert pas à plus que ça ?

Depuis WampServer 3.2.0, cela ne sert plus qu'à cela, la solution actuelle est mentionnée dans la section suivante sur les hôtes virtuels. Rassurez-vous, si elle est un peu plus compliquée à comprendre et à mettre en place, elle résout aussi bien d'autres cas. 🍊

En attendant, vous pouvez toujours accéder à vos scripts en tapant l'entier de l'URL dans la barre d'adresse.

---

1. Plus d'informations à propos de cet identificateur dans [la documentation officielle](#) ↗



### WampServer avant 3.2.0

Euh, moi j'ai bien des liens, mais quand je clique dessus, je reçois un message d'erreur comme quoi la page ne peut être trouvée!

Oui, c'est connu avant WampServer 3.2.0. Notez cependant que ce n'est pas une page d'erreur 404: **c'est le navigateur qui ne trouve pas le serveur**, ou, plus précisément, la résolution de l'hôte a échoué.



Hé, ho, on vient de l'installer, le serveur! Il fonctionne, sinon on ne pourrait pas accéder à `http://localhost/`, non?

Oui une fois de plus ! 🍊

Mais regardez un peu mieux dans la barre d'adresse quand vous êtes sur l'erreur.

Comme nous vous l'avons dit, pour accéder à un des sites sur lesquels vous travaillez, l'adresse doit être de type `http://localhost/monPremierSite/monFichier.php`. Or là, vous n'avez que `http://monPremierSite/`. Ce n'est pas la même chose, c'est donc normal que cela ne fonctionne pas! Rassurez-vous, il n'y a pas besoin de changer grand chose pour que cela soit correct. 🍊



Tu nous as dit qu'on devait accéder à nos fichier avec des adresses ressemblant à `http://localhost/monPremierSite/monFichier.php`, mais pour `http://localhost/`, on n'a pas de nom de fichier. Pourquoi est-ce que quelque chose s'affiche alors?

Parce que notre serveur est configuré pour afficher des pages "par défaut".

Vous avez déjà compris que dans `http://localhost/monPremierSite/monFichier.php`, on a `monPremierSite` qui est le nom du dossier de votre site, puis on a `monfichier.php`, qui est le nom du fichier dans ce dossier. Mais que se passe-t-il si vous ne donnez pas de nom de fichier?

Dans ce cas, le serveur va chercher s'il y a un fichier `index.html` dans le dossier.

- S'il le trouve, il sera utilisé
- S'il n'y a pas de `index.html`, le serveur va chercher un fichier `index.php`
- S'il le trouve, il sera utilisé
- S'il ne trouve aucun de ces deux fichiers, vous recevrez une erreur HTTP 404, vous disant que le fichier n'a pas été trouvé.


Donc si on n'a pas de fichier spécifié dans l'adresse et qu'il n'y a pas d'erreur, c'est très probablement le contenu de `index.php` ou de `index.html` qui s'affiche. 🍊

Le système peut être un peu plus complexe que ça, c'est surtout pour vous expliquer le principe.

### II.1.2.2. Corriger les liens vers vos projets

Vous l'aurez peut-être remarqué si vous avez fait une mise à jour: les éléments listés sous «Vos projets» sur la page d'accueil de WAMP ne sont plus des liens sur lesquels on peut cliquer. Si vraiment vous ne pouvez pas vous passer de cela, il vous reste tout de même la possibilité de changer ce comportement.



Il est de moins en moins conseillé de faire cette manipulation. Utilisez plutôt [les hôtes virtuels](#) .

Pour ça, il n'y a pas grand-chose à faire avec **WAMP 3.0**. En effectuant un clic droit sur l'icône dans la barre de statut, vous verrez un élément de menu *Paramètres Wamp*. Pointez dessus pour faire s'afficher une série de paramètres qui pourront s'activer et/ou se désactiver au gré de vos envies. L'élément qui nous intéresse ici est celui nommé *Paramètres Wamp*. Le dernier élément listé dans la nouvelle liste qui s'affiche, *Ajouter localhost dans url*, est celui qu'il nous faut : cliquez dessus. L'icône va passer à l'orange brièvement, puis redevenir verte. Rechargez maintenant la page d'accueil de WAMP, et cliquez sur un de vos projets. 🍊

Depuis une version plus récente, l'option a été déplacée pour décourager son utilisation. Il faut aller dans *Paramètres WAMP > Nettoyage automatique > Attention risqué ! Uniquement pour expert* et là enfin se trouve l'option *Autoriser liens sur les projets page d'accueil*.

### II.1.2.2.1. Pour WAMP 2.5 (avant 2.5.5)

Si WAMP 3.0 apporte quelques simplifications, il a bien fallu faire sans avant. Et donc, voici comment.

Dans le dossier `www`, il y a un des fichiers `index.php` dont nous avons parlé plus haut. C'est celui qui gère la page quand vous accédez à l'adresse `http://localhost/`, et c'est aussi dans ce fichier qu'est construite la liste des dossiers pour vos projets.

Seulement quelque part, il y a un petit détail qui fait que vous avez `http://monPremierSite/` au lieu de `http://localhost/monPremierSite/`. Si l'on va regarder le code, on trouve ces lignes:

```
329 <?php
330 // récupération des projets
331 $handle=opendir(".");
332 $projectContents = '';
333 while (($file = readdir($handle))!==false)
334 {
335     if (is_dir($file) && !in_array($file,$projectsListIgnore))
336     {
337         //[modif oto] Ajout éventuel de http:// pour éviter le
                 niveau localhost dans les url
338         $projectContents .= '<li><a href="'.($suppress_localhost ?
                 'http://' : '').$file.'">'.$file.'</a></li>';
339     }
340 }
341 closedir($handle);
342 if (empty($projectContents))
343     $projectContents =
                 "<li>".$langues[$langue]['txtNoProjet']."</li>\n";;
```

Listing 3 – Extrait du script `index.php` fourni avec WAMP 2.5.0. Cette partie montre comment est générée la liste des projets

La ligne mise en évidence contient un test pour déterminer s'il faut ajouter ou non le schéma

## II. Utiliser son serveur

d'URL (la partie `http://`). Par défaut, ce test ajoute le schéma, ce qui nous pose problème. Si vous avez regardé la ligne attentivement, le test se base sur une variable `$suppress_localhost`, qui contient un booléen. Cette variable est définie plus haut dans le script, à la ligne 30.

```
29 <?php
30 $suppress_localhost = true;
```

Listing 4 – Initialisation de la variable `$suppress_localhost` dans `index.php` fourni avec WAMP 2.5

Comme vous le constatez, la valeur de la variable ne vient pas d'un test complexe. Du coup, vous pouvez simplement remplacer `true` par `false`.

Rechargez ensuite `http://localhost/` et cliquez sur l'un de vos projets listés. Vous pouvez désormais y accéder normalement. 🎉

Maintenant que vous savez comment organiser vos fichiers et comment y accéder, sachez cependant que la modification effectuée dans `index.php` n'est pas la meilleure des solutions. Cela fonctionnera pour de petits sites simples, mais dès que vous passerez à la vitesse supérieure ou que vous souhaitez utiliser un framework, vous risquez d'avoir des problèmes.

Nous allons donc voir comment mieux organiser notre "garde-manger" par la suite. 🍌

### II.1.3. Les <VirtualHost>

Maintenant que vous avez créé des projets, vous souhaitez faire "plus professionnel" et pouvoir vous affranchir de la partie `http://localhost/` qui apparaît dans l'adresse, et utiliser une URL comme `http://mon-projet.localhost`.

Ou alors, vous utilisez un framework, et vous vous rendez compte qu'il ne fonctionne pas très bien, `http://localhost/` réapparaît dans les adresses générées.

Ceci est souvent dû à l'utilisation d'une variable de PHP qui donne différentes informations à propos du serveur: `$_SERVER`, dont vous pourrez trouver la documentation avec [ce lien ↗](#). Or, il se trouve que cette variable est un tableau où se trouve (entre autres) le nom de l'hôte virtuel ou l'adresse utilisée pour accéder à un script. Et sans changement de votre part, ces valeurs contiennent `localhost` pour tous vos projets. C'est afin de pallier cela qu'il est vivement recommandé d'utiliser les **hôtes virtuels**.

#### II.1.3.1. L'intérêt des hôtes virtuels

Les hôtes virtuels (connus en anglais sous le nom de **virtual hosts** ou, souvent vu à l'écrit, **vhosts**) vont nous permettre de lier un nom de domaine à l'un de nos projets contenus dans le répertoire `www` de WAMP. Ainsi, le projet `http://localhost/monPremierSite` pourra être accessible via l'URL `http://premier-site.localhost`, de même que le projet `http://localhost/monSecondSite` pourra être accessible via l'URL `http://second-site.localhost`.

L'intérêt des virtual hosts est double:

- raccourcir de manière significative l'URL d'accès à votre projet
- s'affranchir des URLs relatives et absolues qui vous poseront problème lors du déploiement du projet en production

---

1. `%WAMPDIR%` est à remplacer par le chemin d'installation de WAMP

## II. Utiliser son serveur

Si nous reprenons notre garde-manger, vous n'allez pas simplement tout mettre dedans et devoir fouiller l'entier pour trouver ce dont vous avez besoin: vous allez avoir différentes armoires qui contiendront diverses choses afin de les retrouver plus facilement. Ici, c'est un peu différent: finalement, on ne va pas trier les ingrédients par types, mais par plats! 🍊

Un hôte virtuel permet donc de regrouper les fichiers par projet (comme le dossier créé dans `www`, cela reste nécessaire), mais en plus vous pourrez y accéder directement avec une adresse plus courte – comme si vous saviez dans quelle partie de votre garde-manger vous pourriez trouver *tous* les ingrédients nécessaires à un plat précis, même ceux qui sont communs à tous vos plats. 🍊

?

Puis-je utiliser n'importe quel nom de domaine?

Oui, absolument n'importe quel nom de domaine, même si:

1. l'utilisation d'un nom de domaine existant ou "final" est **déconseillée**:
  - vous ne pourriez plus accéder au vrai contenu sans changer en permanence des paramètres;
  - certains navigateurs remarquent que le nom de domaine pointe vers une adresse IP inhabituelle et vous demandent de confirmer l'accès.
2. certains noms de domaine forcent le HTTPS (notamment tous ceux en `.dev` avec les navigateurs actuels).

Pour illustrer le premier cas, imaginons que vous souhaitiez utiliser `google.ch`, par exemple. Après toutes les étapes suivies pour paramétrer cet hôte virtuel sur votre machine, vous ne pourriez plus accéder à sa version internet.

Allons dans le plus problématique: vous savez que vous allez publier votre projet au domaine final `mon-projet.com`. Développer en local avec le domaine `mon-projet.com` comme hôte virtuel risque de vous faire confondre production et développement quand vous testerez sur le serveur de production. Vous voyez le problème venir, non? Sauf si vous êtes toujours exactement à votre affaire (et dans ce cas, nous vous tirons notre chapeau), vous risquez:

- de vérifier vos corrections en local sur la version de production, donc vous ne les verrez pas et ne comprendrez pas immédiatement pourquoi
- de vérifier si vos corrections ont bien été poussées en production alors que vous visionnez votre version locale. Vous les verrez, pas le client, et là non plus, vous ne comprendrez pas tout de suite la raison, et c'est un peu plus gênant quand le commanditaire est en train de s'énerver. 🍊

Donc **évitez d'utiliser un nom de domaine existant ou "final"/"de production" quand vous développez**. Cela vous aidera à vous y retrouver, vous évitera d'avoir à confirmer à chaque accès, et vous évitera des situations embarrassantes. 🍊

Quant au second cas, c'est plus gênant. Même si vous pouvez évidemment paramétrer votre WAMP pour qu'il puisse répondre à des requêtes HTTPS, les navigateurs vont vous faire remarquer que le certificat est auto-signé, et risquent d'afficher des messages à ce propos...

Voici de quoi vous aider. De même que vous pouvez choisir le nom de domaine, c'est à vous de choisir le **TLD**<sup>1</sup>. Ici, afin d'éviter les soucis mentionnés plus haut, nous vous proposons ceci: nous utiliserons le **TLD** `.localhost`. Cela nous permettra de différencier vos projets en local de vos projets en production:

- `mon-projet.fr` en production
- `mon-projet.localhost...` en local 🍊

Ce **TLD** est expressément réservé pour une utilisation sur la même machine—ça tombe bien,

## II. Utiliser son serveur

c'est exactement notre cas —, et cette réservation a été effectuée par l'organisme qui gère les TLDs. Donc il n'y quasiment aucun risque de le voir disparaître ou devenir gênant!

Notez au passage que vous n'avez pas besoin de nommer les URLs comme les dossiers de vos projets, ce serait plutôt l'inverse: pour mieux vous retrouver dans vos projets quand vous y accédez par l'explorateur Windows, vous aurez probablement utilisé le nom de domaine de votre projet comme nom de dossier. 🍊

### II.1.3.2. Edition du fichier `hosts` de Windows

i

Le fichier `hosts` est un **fichier caché du système**. Pour pouvoir le trouver, il vous faut donc afficher ces fichiers.

Rendez-vous dans le Panneau de Configuration, rubrique *Option des dossiers*. Dans la fenêtre qui s'ouvre alors, cliquez sur l'onglet *Affichage*, et cochez le bouton radio *Afficher les dossiers les fichiers, dossiers et lecteurs cachés*, puis cliquez sur *Appliquer*.

Vous pourrez recocher *Ne pas afficher les fichiers, dossiers ou lecteurs cachés* une fois les modifications effectuées.

Sur Windows comme sur UNIX, il existe un fichier nommé `hosts` qui remplit plus ou moins le même rôle qu'un **serveur DNS** 🌐 : il va préciser quelle adresse IP est associée à un nom de domaine. Ce fichier est consulté avant tout accès à un serveur DNS externe. C'est dans ce fichier que vous allez renseigner vos noms de domaines personnalisés.

Afin de pouvoir enregistrer les modifications, ouvrez votre éditeur de texte favori **en mode Administrateur**, soit avec le traditionnel clic droit > *Exécuter en tant qu'administrateur*. Si vous ne le faites pas, vous ne pourrez pas enregistrer les modifications (vous recevrez un message d'erreur, ou il vous sera demandé de vérifier si le fichier n'est pas utilisé), qui seront donc inutiles.

Maintenant que votre éditeur de texte est correctement lancé, utilisez le menu de celui-ci pour ouvrir le fichier `hosts` situé à l'emplacement `C:\Windows\System32\drivers\etc`.

Ajoutez les lignes suivantes en bas du fichier (une règle par ligne):

```
1 <?php
2 $suppress_localhost = true;
```

Listing 5 – Lignes à ajouter dans le fichier `hosts`

Enregistrez et fermez.

Afin de s'assurer que Windows ait bien pris en compte les modifications qu'on vient d'ajouter au fichier `hosts`, ouvrez l'invite de commande et lancez `ipconfig /flushdns`. Si vous comprenez un peu l'anglais, vous devinerez que ceci va purger les définitions DNS que Windows a enregistrées. Du coup, le fichier `hosts` sera à nouveau lu, donc saura que les domaines `premier-site.localhost` et `second-site.localhost` se trouvent à l'adresse `127.0.0.1 (:::1)`. 🍊

Dès lors, lorsque vous tenterez d'accéder à `http://premier-site.localhost` ce fichier indiquera qu'il faut s'adresser à la machine portant l'IPv4 `127.0.0.1` (ou l'IPv6 correspondante `:::1`), qui n'est autre que votre WAMP.



Vous pourrez trouver sur différents sites qu'il faut ajouter ou décommenter les lignes

```
1 <?php
2 $suppress_localhost = true;
```

Listing 6 – Indique que `localhost` correspond aux adresses IPv4 `127.0.0.1` et IPv6 `::1`

Ceci n'est **plus du tout nécessaire dès Windows 7<sup>2</sup>, 3<sup>3</sup>**, mais reste donc à faire pour Windows Vista ou moins récent.

Sachez néanmoins que, WAMP étant prévu pour fonctionner sur plusieurs versions de l'OS, l'application continue à effectuer cette opération même s'il n'y en a plus besoin. Ne vous étonnez donc pas de découvrir ces lignes dé-commentées après avoir utilisé WAMP.



Si vous souhaitez rendre votre site accessible avec les deux adresses `http://www.premier-site.localhost` et `http://premier-site.localhost`, il faut aussi renseigner ces noms dans le fichier `hosts`.

En fait, dès qu'on souhaite utiliser un *sous-domaine*, que ce soit `www` ou `dev` ou ce que vous voulez qui soit avant votre nom de domaine proprement dit (ici `premier-site.localhost`), il faut aussi le renseigner dans le fichier `hosts`. Mais on peut le faire très simplement, sans ajouter de ligne, en ajoutant juste le nouveau nom de domaine complet sur la même ligne. Reprenons l'exemple et adaptons-le pour le premier nom de domaine. Nous devrions avoir:

```
1 <?php
2 $suppress_localhost = true;
```

Listing 7 – Ajout de la gestion DNS du sous-domaine `www` pour `premier-site.localhost`. Faites quand même attention au fait que vous ne pouvez mettre ainsi que jusqu'à 8 noms sur une même ligne, pas plus! 🍊

### II.1.3.3. Définition des hôtes virtuels



Un instant... mes deux URLs personnalisées pointent toutes deux vers la même IP. Comment pourrais-je accéder au bon projet?

Voilà la question que nous attendions! 🍊

Il y a justement un fichier de configuration Apache qui va permettre de router les bonnes URLs vers les bons répertoires. Ce fichier se trouve ici: `%APACHEDIR%\conf\extra\httpd-vhosts.conf4`. Éditez ce fichier, qui doit ressembler à cela:



## II. Utiliser son serveur

```
1 #
2 # Virtual Hosts
3 #
4 # If you want to maintain multiple domains/hostnames on your
5 # machine you can setup VirtualHost containers for them. Most
6 # configurations
7 # use only name-based virtual hosts so the server doesn't need to
8 # worry about
9 # IP addresses. This is indicated by the asterisks in the
10 # directives below.
11 #
12 # Please see the documentation at
13 # <URL:http://httpd.apache.org/docs/2.2/vhosts/>
14 # for further details before you try to setup virtual hosts.
15 #
16 # You may use the command line option '-S' to verify your virtual
17 # host
18 # configuration.
19 #
20 # Use name-based virtual hosting.
21 #
22 # NameVirtualHost *:80
23 #
24 # VirtualHost example:
25 # Almost any Apache directive may go into a VirtualHost container.
26 # The first VirtualHost section is used for all requests that do
27 # not
28 # match a ServerName or ServerAlias in any <VirtualHost> block.
29 #
30 <VirtualHost *:80>
31     ServerAdmin webmaster@dummy-host.example.com
32     DocumentRoot "C:/Apache2/docs/dummy-host.example.com"
33     ServerName dummy-host.example.com
34     ServerAlias www.dummy-host.example.com
35     ErrorLog "logs/dummy-host.example.com-error.log"
36     CustomLog "logs/dummy-host.example.com-access.log" common
37 </VirtualHost>
38
39 <VirtualHost *:80>
40     ServerAdmin webmaster@dummy-host2.example.com
41     DocumentRoot "C:/Apache2/docs/dummy-host2.example.com"
42     ServerName dummy-host2.example.com
43     ErrorLog "logs/dummy-host2.example.com-error.log"
44     CustomLog "logs/dummy-host2.example.com-access.log" common
45 </VirtualHost>
```

Listing 8 – Contenu initial du fichier %APACHEDIR%\conf\extra\httpd-vhosts.conf<sup>5</sup>



## II. Utiliser son serveur

La partie haute du fichier est composée de commentaires (lignes précédées par des #), la partie basse contient deux définitions d'hôtes virtuels donnés à titres d'exemple. Chaque hôte virtuel est défini dans le bloc `<VirtualHost *:80> </VirtualHost>`. Nous n'aurons pas besoin de ces exemples, vous pourrez donc les supprimer.

Pour que votre projet `http://localhost/monPremierSite` soit accessible à l'URL `http://premier-site.local`, créez le bloc de règles ci-dessous à la suite dans le fichier:

```
1 <VirtualHost *:80>
2     ServerAdmin votreemail@exemple.com
3     DocumentRoot "C:/wamp/www/monPremierSite"
4     ServerName premier-site.localhost
5     ServerAlias www.premier-site.localhost
6     ErrorLog "logs/premier-site.localhost-error.log"
7     CustomLog "logs/premier-site.localhost-access.log" common
8 </VirtualHost>
```

Listing 9 – Définition d'hôte virtuel liant le nom de domaine `premier-site.local` au dossier `C:\wamp\www\monPremierSite\`

i

C'est fait... mais je n'ai rien compris! 🤔

Une petite explication s'impose.

- nous avons créé une nouvelle règle, définie dans le bloc `<VirtualHost *:80> </VirtualHost>`. Tout ce qui se trouve dans ce bloc ne concernera que l'hôte virtuel ainsi défini
- `ServerAdmin` indique l'adresse email de l'administrateur de l'hôte virtuel (vous, en l'occurrence!). Si un jour une erreur du serveur survient, le visiteur aura un message l'invitant à contacter cette adresse email;
- `DocumentRoot` précise vers quel dossier doit pointer le nom de domaine défini à la ligne suivante, **sans slash / ni antislash \ final**
- `ServerName` indique le nom de domaine pour l'hôte virtuel, **sans sous-domaine**
- `ServerAlias` indique une URL alternative  
Ici, on permet la prise en compte des sous-domaines. Dans l'exemple, c'est le sous-domaine par défaut `www` qui est pris en compte
- `ErrorLog` indique le chemin vers le fichier journal où seront consignées toutes les erreurs PHP rencontrées par la suite
- `CustomLog` indique le chemin vers le fichier journal où seront consignés tous les accès HTTP

Faisons de même pour le second projet:

```
1 <VirtualHost *:80>
2     ServerAdmin votreemail@exemple.com
3     DocumentRoot "C:/wamp/www/monSecondSite"
4     ServerName second-site.localhost
5     ServerAlias www.second-site.localhost
6     ErrorLog "logs/second-site.localhost-error.log"
```

## II. Utiliser son serveur

```
7     CustomLog "logs/second-site.localhost-access.log" common
8 </VirtualHost>
```

Listing 10 – Définition d’hôte virtuel liant le nom de domaine `second-site.localhost` au dossier `C:\wamp\www\monSecondSite\`

Enregistrez et fermez ce fichier.

Pour que vos virtual hosts soient fonctionnels, il ne reste plus qu’à indiquer à WAMP de prendre en compte ce fichier. Pour cela, éditez le fichier `httpd.conf` situé ici `%APACHE DIR%\conf\httpd.conf`.

Quelque part vers la fin du fichier (les numéros de ligne peuvent varier d’une version à l’autre de WAMP), vous trouverez ceci

```
512 # Virtual hosts
513 #Include conf/extra/httpd-vhosts.conf
```

Listing 11 – Inclusion de la configuration relative aux hôtes virtuels dans `httpd.conf`

Décommentez la ligne `#Include conf/extra/httpd-vhosts.conf` en enlevant le signe `#` du début de ligne, puis redémarrez WAMP.

Maintenant, si tout s’est bien passé, quand vous entrez l’adresse `http://premier-site.localhost` ou `http://www.second-site.localhost`, vous arrivez sur la page d’accueil des projets respectifs.

*i*

Depuis WAMP 3.0.7, il y a un outil disponible depuis la page d’accueil de WAMP qui permet de se faciliter la vie. Et depuis la version 3.2.8, il est même devenu possible de spécifier des versions de PHP spécifiques à chaque hôte virtuel ainsi créé.

Par contre, il n’est possible de spécifier qu’un unique nom d’hôte, vous ne pourrez paramétrer par ce biais que l’un ou l’autre `premier-site.localhost` ou `www.premier-site.localhost`. Pour celui qui n’a pas été saisi, il vous faudra aller ajouter `ServerAlias` dans la définition, ainsi que la correspondance vers l’adresse IP dans le fichier `hosts`

### II.1.3.4. Conserver l’accès à `http://localhost/` sur certaines versions de WAMP

Il y a un petit désavantage à créer ces hôtes virtuels: suivant votre version de WAMP, quand vous essayez d’accéder à `http://localhost/`, vous arrivez sur la page d’accueil de votre premier site!

En fait, c’est parce qu’Apache redirige toutes les requêtes qui arrivent sur la machine (et qui ne sont pas destinées à un hôte virtuel existant) vers le premier hôte virtuel accessible défini, et s’il n’y en a pas, vers la racine serveur (notre fameux dossier `www`). Or, il peut arriver que, pour des tests simples par exemple, vous ne souhaitiez pas les faire dans un hôte virtuel. `localhost` serait tout indiqué, mais vous n’y avez plus accès...

?

Hé! Vous venez de nous dire que les requêtes qui arrivent sur la machine aboutissent par défaut dans le premier hôte virtuel. Pourquoi ne pas créer un hôte virtuel pour `localhost`, alors?

Vous voyez, vous avez compris tout seuls! 🍊

C'est exactement ce que l'on va faire: créer un hôte virtuel pour `localhost`. Afin qu'il soit vraiment l'hôte virtuel par défaut, il faut qu'il soit placé avant tous les autres dans le fichier `httpd-vhosts.conf`.

Notez qu'on peut aussi beaucoup simplifier sa déclaration, qui pourra se résumer à ceci:

```
1 <VirtualHost *:80>
2     ServerName localhost
3     ServerAlias localhost
4     DocumentRoot "%WAMPDIR%/www"
5     <Directory "%WAMPDIR%/www/">
6         Options +Indexes +Includes +FollowSymLinks
7             +MultiViews
8         AllowOverride All
9         Require local
10    </Directory>
11 </VirtualHost>
```

Listing 12 – Définition de l'hôte virtuel pour `localhost`. Remplacez `%WAMPDIR%` par le chemin de votre installation de WAMP

### II.1.4. PHP en ligne de commandes

Les frameworks actuels font volontiers usage de la ligne de commandes pour lancer des scripts PHP. Seulement, WAMP ne fait pas de réglage à ce niveau, car il est initialement prévu pour une utilisation avec Apache. Il nous faut donc pallier cela. 🍊

Rendez-vous d'abord dans le panneau de configuration, puis la rubrique Système (sous Windows 7 et plus récent, vous pouvez utiliser le raccourci

; +  **Pause**). Dans la partie de gauche de la fenêtre, vous trouverez une rubrique *Paramètres système avancés*, pour laquelle Windows vous demandera d'être administrateur pour y accéder. S'ouvre alors une fenêtre avec une série d'onglets. Si ce n'est pas celui qui est déjà choisi, cliquez sur *Paramètres système avancés* une fois de plus. Dans cet onglet, en bas à droite, se trouve un bouton *Variables d'environnement*. Cliquez dessus pour ouvrir encore une nouvelle fenêtre.

---

1. Ce sont les quelques dernières lettres entre le dernier point et le slash suivant dans une URL, comme `.com`, `.fr` ou `.`. Plus d'informations sur [la page Wikipédia consacrée](#) [↗](#)

2. <http://support.microsoft.com/kb/972034/fr#LetMeFixItMyselfAlways> [↗](#), partie "Windows 7 et versions antérieures de Windows", 3e exemple de code, dernières lignes (dernier accès le 24 août 2014)

3. [http://answers.microsoft.com/en-us/windows/forum/windows\\_8-networking/how-can-i-reset-ms-host-files-in-windows-8/4057335b-6160-4073-aafb-4412db9db971](http://answers.microsoft.com/en-us/windows/forum/windows_8-networking/how-can-i-reset-ms-host-files-in-windows-8/4057335b-6160-4073-aafb-4412db9db971) [↗](#), réponse d'Anshul Khanna du 31 décembre 2012, partie "Méthode 2", point 7, deux dernières lignes de contenu mentionnées (dernier accès le 24 août 2014)

4. `%APACHEDIR%` est à remplacer par le chemin de l'installation d'Apache

5. Idem

## II. Utiliser son serveur

Cette dernière est séparée en deux parties: les variables pour l'utilisateur en haut, et les variables système, partagées entre tous les utilisateurs, en dessous. C'est parmi celles-ci qu'il vous faut trouver `Path` et la sélectionner. Une fois ceci fait, cliquez sur *Modifier...* Encore une popup qui s'ouvre—ce n'est pas pour rien que le système d'exploitation s'appelle Windows. 🍊  
A partir de là, la suite dépend de votre version de Windows et surtout de ce que contient cette nouvelle fenêtre.

- **La fenêtre comporte une série de lignes à gauche, et des boutons à droite**

👁️ Contenu masqué n°1

- **La fenêtre comporte deux champs texte l'un sous l'autre**

👁️ Contenu masqué n°2

Ensuite de quoi, fermez progressivement les diverses fenêtres en cliquant sur *Appliquer* quand c'est possible, sinon sur *OK*.

Vous devriez pouvoir utiliser PHP en ligne de commandes désormais. Ouvrez l'invite de commandes et tapez `php -v`: le numéro de la version de PHP devrait s'afficher alors.

*i*

Si vous aviez déjà l'invite de commandes ouverte quand vous aviez modifié les variables d'environnement, il vous faut la fermer et la rouvrir. Les programmes qui sont en cours d'exécution lors de ce genre de modifications ne peuvent en tenir compte.

!

La ligne de commandes n'utilise pas le même fichier `php.ini` que celui pour Apache. Or, c'est ce dernier qui est modifié quand vous utilisez le menu de WAMP pour faire la configuration de PHP.

Il vous faudra comparer les deux fichiers pour activer les mêmes extensions et paramétrer les mêmes valeurs (fuseau horaire, temps maximal d'exécution, limite de mémoire allouée, etc.).

Il est cependant à noter qu'il doit y avoir des différences entre les deux fichiers de configuration: la ligne de commande utilise des points que le module Apache ne sert pas, et vice-versa. Si vous ne savez pas quoi faire de certaines différences, laissez-les, c'est plus sage.

*i*

Depuis la version 3.1.2, un message s'affiche quand une version de PHP que WAMP peut utiliser est renseignée pour la ligne de commandes. Ceci est en prévision d'un potentiel conflit de configuration entre la version utilisée en interne par WAMP et celle de la ligne de commandes<sup>5</sup>.

Pour désactiver ce message, effectuez un clic droit sur l'icône WAMP dans la barre de statut, puis pointez *> Paramètres Wamp* et cliquez sur *Ne pas vérifier PATH*.

### II.1.4.0.1. cURL et HTTPS

Si vous avez besoin de PHP en ligne de commandes, c'est probablement parce que vous allez utiliser composer. Vous avez peut-être déjà tenté de lancer une commande, et vous vous retrouvez avec le message suivant.

```
1 cURL error 60: SSL certificate problem: unable to get local issuer certificate
```

#### Listing 13 – Erreur quand cURL ne peut pas vérifier une connexion sécurisée

En fait, le problème est que cURL tente de se connecter quelque part en HTTPS, mais qu'il n'arrive pas à vérifier l'authenticité du certificat pour la connexion sécurisée. Pour que cela lui soit possible, il faut lui fournir une base de données qui lui permettra de faire la validation.

Dans un premier temps, il faut télécharger cette base de données, qui se présente sous la forme d'un seul fichier. Vous pouvez le récupérer à l'adresse <https://curl.haxx.se/ca/cacert.pem>. Enregistrez-le quelque part où vous ne risquez pas de l'effacer, nous vous proposons de créer un dossier `%WAMPDIR%\ssl`<sup>6</sup> pour cela.

Ensuite, ouvrez le fichier `php.ini` avec votre éditeur de texte favori, et recherchez les lignes mentionnées ci-dessous.

```
1 [curl]
2 ; A default value for the CURLOPT_CAINFO option. This is required
   to be an
3 ; absolute path.
4 ;curl.cainfo =
```

#### Listing 14 – La partie de `php.ini` qui permet de renseigner la base de données

Dé-commentez la dernière ligne, et ajoutez à la fin de celle-ci le chemin du fichier que vous avez téléchargé. C'est prêt! 🎉

### II.1.5. Accéder à votre serveur depuis une machine distante

Quand on a un serveur de développement, on souhaite parfois pouvoir utiliser celui-ci comme un hébergeur, au moins pour les petits projets, avant de les rendre publics. WAMP permet plus ou moins de jouer ce rôle, mais cela demande quelques paramètres suivant ce que vous souhaitez rendre accessible.



Par défaut, la racine de votre serveur (le dossier `www`, qui contient la page d'accueil de WAMP), n'est accessible que depuis votre propre machine, et c'est recommandé de laisser cela ainsi. Il est vivement conseillé d'utiliser des [hôtes virtuels](#) si vous souhaitez partager

1. `%WAMPDIR%` est à remplacer par le chemin de votre installation de WAMP
2. `%PHP_VERSION%` est à remplacer par le numéro de version complet de PHP, avec les points
3. `%WAMPDIR%` est à remplacer par le chemin de votre installation de WAMP
4. `%PHP_VERSION%` est à remplacer par le numéro de version complet de PHP, avec les points
5. Pour plus d'information, voir [ce sujet sur le forum officiel](#)
6. `%WAMPDIR%` est à remplacer par le chemin de votre installation de WAMP



l'accès à votre travail.



Cette partie dépend fortement de réglages de votre routeur/box Internet qu'il nous est impossible de traiter ici. Il vous faudra certainement vous renseigner sur ses possibilités.

### II.1.5.1. S'ouvrir au monde

Quand vous lancez WAMP pour la première fois, il est prévu pour ne laisser accéder que les demandes émanant de votre machine. Toute personne qui tenterait depuis un client quelconque d'accéder à l'un des sites qui se trouvent sur votre ordinateur recevrait une erreur 403.

Initialement, pour permettre à d'autres machines d'accéder à vos sites, il y avait un élément de menu qui permettait de dire *Passer en ligne* ou, à l'inverse, *Passer hors ligne*. Cette option est masquée par défaut avec les versions de WAMP 3.1. Qui plus est, si vous avez créé des hôtes virtuels, cette commande ne changera rien pour ceux-ci. Du coup, il faudra aller modifier quelques lignes de configuration armé de votre éditeur de texte favori.

Rendez-vous dans le fichier qui contient les configurations de vos hôtes virtuels, soit `%WAMP_DIR%\bin\apache\apache%APACHE_VERSION%\conf\extra\httpd-vhosts.conf1, 2`.

Pour l'hôte virtuel que vous souhaitez rendre accessible, recherchez la ligne `Require local`, et remplacez par `Require all granted`. Redémarrez ensuite Apache. Si ça redevient vert, cette première étape est terminée ! 🎉

### II.1.5.2. Rediriger les gens chez vous

Pour l'instant, vous avez ouvert la porte sur votre ordinateur. Il faut encore renseigner les autres ordinateurs de l'endroit où elle est.

#### II.1.5.2.1. Accessibilité sur le LAN<sup>5</sup>

Sur Internet, pour pouvoir afficher un site, il y a plusieurs requêtes qui sont effectuées, afin de connaître l'adresse IP de la machine où le site est hébergé. Cela implique le passage par des **DNS**. En conséquence, si vous souhaitez accéder à votre serveur depuis une autre machine, il vous faut soit un serveur **DNS**, soit utiliser directement l'adresse IP de votre machine.

Vous pouvez éventuellement faire une recherche «quelle est mon adresse IP» et utiliser celle qui vous est donnée. Mais faire ainsi va passer par le **WAN**<sup>6</sup>, alors que vous vous trouvez sur le même **LAN**. Vous n'avez pas besoin de passer par une machine qui peut se trouver ailleurs sur le continent pour trouver un appareil qui se trouve à maximum 20 mètres de vous, n'est-ce pas?

Une solution serait d'utiliser l'adresse IP *locale* de la machine qui fait office de serveur. Mais on se retrouverait avec un problème similaire à celui qui nous empêchait d'accéder à localhost dans la partie des hôtes virtuels: cette méthode ne permet d'accéder qu'au premier vhost défini sur le serveur. Notons au passage que le problème surviendrait avec l'IP publique aussi.

Une autre solution (probablement la plus pratique) serait d'avoir un serveur **DNS** installé sur votre routeur/box Internet. Certains modèles le proposent, à vous de vous renseigner à ce sujet. Maintenant, si vous avez encore en tête la solution que nous avons mise en place pour nos hôtes virtuels, vous avez déjà compris que nous allons utiliser le fichier `hosts` des clients de votre **LAN**, ce fichier qui fait office de **DNS** "prioritaire".

Pour ce faire, il nous faut dans un premier temps l'adresse locale du serveur. Ouvrez l'invite de

## II. Utiliser son serveur

commandes et lancez `ipconfig`. Vous aurez des informations sur les cartes réseau existantes sur votre machine, et notamment (pour celle qui sert à la connexion) deux lignes nommées `Adresse IPv6` et `Adresse IPv4`, comme dans l'exemple ci-dessous:

```
1 Carte Ethernet Connexion au réseau local :
2
3   Suffixe DNS propre à la connexion. . . : ethernet
4   Adresse IPv6. . . . . :
5   fd00::f856:66de:b181:c036
6   Adresse IPv6 temporaire . . . . . :
7   fd00::24ed:1051:ea57:7bce
8   Adresse IPv6 de liaison locale. . . . . :
9   fe80::f855:66de:b181:c036%15
10  Adresse IPv4. . . . . : 192.168.0.2
11  Masque de sous-réseau. . . . . : 255.255.255.0
12  Passerelle par défaut. . . . . : 192.168.0.1
13
14 Carte réseau sans fil Connexion au réseau sans fil :
15
16  Statut du média. . . . . : Média déconnecté
17  Suffixe DNS propre à la connexion. . . : wifi
```

Listing 15 – Résultat d'un appel à `ipconfig` avec une connexion par câble Ethernet

Ce sont les deux adresses à utiliser dans le fichier `hosts`, avec les noms de domaine que vous utilisez sur celle-ci. Pour faire court, en reprenant les divers exemples utilisés dans ce tutoriel, vous pourrez ajouter ces lignes dans les fichiers `hosts` des clients:

```
1 Carte Ethernet Connexion au réseau local :
2
3   Suffixe DNS propre à la connexion. . . : ethernet
4   Adresse IPv6. . . . . :
5   fd00::f856:66de:b181:c036
6   Adresse IPv6 temporaire . . . . . :
7   fd00::24ed:1051:ea57:7bce
8   Adresse IPv6 de liaison locale. . . . . :
9   fe80::f855:66de:b181:c036%15
10  Adresse IPv4. . . . . : 192.168.0.2
11  Masque de sous-réseau. . . . . : 255.255.255.0
12  Passerelle par défaut. . . . . : 192.168.0.1
13
14 Carte réseau sans fil Connexion au réseau sans fil :
15
16  Statut du média. . . . . : Média déconnecté
17  Suffixe DNS propre à la connexion. . . : wifi
```

Listing 16 – Déclaration des noms de domaine dans les fichiers `hosts` des clients

Si vous avez bien suivi, vous pourrez avec ça utiliser l'adresse `http://premier-site.local` sur un client et cela affichera bien le site demandé.

### II.1.5.2.2. Laisser accéder à phpMyAdmin

Par défaut, phpMyAdmin n'accepte pas non plus les connexions depuis l'extérieur. Mais il ne suffit pas de passer en ligne pour régler cela: il y a une consigne à changer dans la définition de l'alias pour phpMyAdmin.

```
1 Alias /phpmyadmin "%WAMPDIR%/apps/phpmyadmin%PMA_VERSION%/"
2
3 # to give access to phpmyadmin from outside
4 # replace the lines
5 #
6 # Require local
7 #
8 # by
9 #
10 # Require all granted
11 #
12
13 <Directory "%WAMPDIR%/apps/phpmyadmin%PMA_VERSION%/">
14     Options Indexes FollowSymLinks MultiViews
15     AllowOverride all
16     <IfDefine APACHE24>
17         Require local
18     </IfDefine>
19     <IfDefine !APACHE24>
20         Order Deny,Allow
21         Deny from all
22         Allow from localhost ::1 127.0.0.1
23     </IfDefine>
24     php_admin_value upload_max_filesize 128M
25     php_admin_value post_max_size 128M
26     php_admin_value max_execution_time 360
27     php_admin_value max_input_time 360
28 </Directory>
```

Listing 17 – Contenu du fichier `%WAMPDIR%\alias\phpmyadmin.conf`<sup>3</sup> fourni avec WAMP.

La version de phpMyAdmin a été remplacée par `%PMA_VERSION%`

Les quatre lignes mises en évidence sont des réglages qui font en sorte que l'on ne peut accéder à phpMyAdmin qu'en étant sur la même machine. Si, pour la première, le changement à effectuer est expliqué plus haut (aux lignes 3 à 11), les trois suivantes sont moins simples à appréhender. En fait, je vous propose d'adapter un petit peu le contenu du fichier pour y faire figurer ceci:

```
1 Alias /phpmyadmin "%WAMPDIR%/apps/phpmyadmin%PMA_VERSION%/"
2
3 # to give access to phpmyadmin from outside
4 # replace the lines
5 #
6 # Require local
```



## II. Utiliser son serveur

```
7 #
8 #   and
9 #
10 # Order Deny,Allow
11 # Deny from all
12 # Allow from localhost ::1 127.0.0.1
13 #
14 #   by respectively
15 #
16 # Require all granted
17 #
18 #   and
19 #
20 # Order Allow,Deny
21 # Allow from all
22 #
23
24 <Directory "%WAMPDIR%/apps/phpmyadmin%PMA_VERSION%/">
25     Options Indexes FollowSymLinks MultiViews
26     AllowOverride all
27     <IfDefine APACHE24>
28         Require local
29     </IfDefine>
30     <IfDefine !APACHE24>
31         Order Deny,Allow
32         Deny from all
33         Allow from localhost ::1 127.0.0.1
34     </IfDefine>
35     php_admin_value upload_max_filesize 128M
36     php_admin_value post_max_size 128M
37     php_admin_value max_execution_time 360
38     php_admin_value max_input_time 360
39 </Directory>
```

Listing 18 – Un nouveau contenu pour `%WAMPDIR%\alias\phpmyadmin.conf`<sup>4</sup> fourni avec WAMP. La version de phpMyAdmin a été remplacée par `%PMA_VERSION%` Et du coup, vous avez toutes les informations nécessaires pour passer en ligne ou restreindre à nouveau l'accès! 🍊



Notez que permettre l'accès distant à phpMyAdmin avec la configuration fournie par WAMP, qui ne demande pas de vous identifier pour y accéder et ne vous force pas à donner un mot de passe pour l'utilisateur root, pose un très gros problème de sécurité. N'oubliez pas que WAMP est avant tout prévu pour une utilisation en amateur ou développement, et cela se limite souvent à être accessible depuis la machine uniquement.

### II.1.5.2.3. Accessibilité depuis n'importe-où

Maintenant que vous avez terminé le site, vous souhaitez que celui pour qui vous l'avez fait puisse y accéder depuis chez lui. On se trouve avec un problème similaire que pour le réseau local, mais il faudra utiliser l'adresse IP globale dans le fichier `hosts` de la machine client. Seulement, il y a en plus le fait que votre box ne va pas nécessairement savoir où est votre serveur. Nous vous expliquons pourquoi.

En local, vous aviez utilisé l'adresse IP locale de votre serveur. Votre boîtier la connaissait, et pouvait ainsi router les requêtes correctement. Mais ces adresses IP locales sont, comme leur dénomination l'indique, locales, donc connues dans le LAN seulement. Votre LAN est connecté au WAN par votre routeur, et c'est lui qui est votre porte vers Internet, c'est lui qui possède l'adresse IP publique. Si vous voulez le vérifier, vous pourrez voir que tous les appareils connectés à Internet ont la même adresse IP publique. Le souci est le suivant: comment faire pour que votre partenaire puisse accéder à votre serveur précisément, un des éléments de votre LAN, en n'ayant que votre adresse IP publique?

Si vous n'avez qu'un seul serveur, vous pouvez probablement rediriger le port utilisé pour HTTP vers votre machine d'hébergement (on parle de **roulage NAT**, ou de *redirection de port*). Ainsi, toutes les requêtes HTTP qui arriveront de l'extérieur arriveront à la bonne machine.

Mais si vous avez plusieurs machines qui servent d'hébergement, on revient au besoin d'un DNS.



La plupart des routeurs distribuent des adresses IP locales dynamiquement à la connexion de l'appareil qui en a besoin, donc vous vous connectez une fois, éteignez l'appareil, le rallumez et vous reconnectez, et les adresses ne seront plus les mêmes.

De plus, cette adresse va changer si vous changez de type de connexion et passez de WiFi à câble sans redémarrer votre serveur.

Vous pouvez regarder les réglages de votre boîtier pour le configurer afin qu'il assigne une adresse IP locale fixe à votre serveur. Mais là encore, selon le routeur et la version du firmware, vous pourrez peut-être assigner une adresse IP

- à une adresse MAC, ce qui vous obligera à vous connecter par WiFi ou par câble, donc ne vous permettra pas de jongler entre les deux
- à un nom de machine, qui vous assignera la même adresse IP quelle que soit votre moyen de connexion



Même si cela tend à se stabiliser, vous pouvez aussi ne pas avoir deux fois la même adresse IP publique. Dans ce cas et si ça pose problème, il vous faudra obligatoirement passer par un DNS.

### II.1.5.3. Quelques obstacles possibles

Il y a quelque chose dont il vous faudra aussi tenir compte: les pare-feux de votre serveur et de votre routeur, ainsi que les antivirus. De manière simple pour voir si l'un ou l'autre pose problème sur le serveur, désactivez le pare-feu et l'antivirus, testez si Apache est accessible, puis testez avec seulement le pare-feu et seulement l'antivirus. Vous vous rendrez compte très rapidement de ce qui pose problème.

Pour l'éventuel pare-feu de votre boîtier, il vous faudra aller regarder dans sa configuration ou son manuel.

## II. Utiliser son serveur

Sous Windows, quand vous avez installé WAMP, votre pare-feu vous a probablement demandé si vous souhaitiez laisser l'accès à Apache sur le réseau local/d'entreprise et/ou sur le réseau public. Référez-vous à la documentation de votre pare-feu pour permettre à Apache (httpd.exe) de discuter avec le monde extérieur.

Parfois, les antivirus n'aiment pas non-plus les programmes qui peuvent se connecter à Internet. Vérifiez dans votre antivirus qu'Apache (httpd.exe) n'est pas bloqué d'une quelconque manière. Vous pourrez probablement ajouter une exception.

Ce sont là toutes les pistes que nous pouvons vous transmettre afin d'ouvrir votre serveur aux accès distants. Faites bien attention aux différents points de sécurité mentionnés, rien n'est anodin. 🍊

## Conclusion

Voilà, vous avez toutes les bases pour vous amuser avec WAMP en amateur. 🍊

Néanmoins, il y a d'autres possibilités intéressantes qui risquent de vous plaire si vous êtes ou vous destinez à entrer dans le milieu du développement web.

## Contenu masqué

### Contenu masqué n°1

Celui qui nous intéresse, c'est `Nouveau`. Après avoir cliqué dessus, le curseur va se placer en bas de liste à gauche, et vous n'aurez qu'à coller le chemin vers l'emplacement de PHP, soit `%WAMPDIR%\bin\php\php%PHP_VERSION%`<sup>1, 2</sup> [Retourner au texte.](#)

### Contenu masqué n°2

On va surtout s'intéresser à celui du bas.



Il ne faut pas changer quoi que ce soit dans le premier champ, sans quoi vous risquez de devoir utiliser un DVD d'installation de votre système d'exploitation pour le réparer, et certains programmes pourraient ne plus fonctionner correctement malgré cela.

On peut voir que ce qu'il contient est une série de chemins, séparés entre eux par des points-virgules ;. Ce sont en fait des chemins de dossiers dans lesquels Windows vérifie, quand lui ou un programme doit lancer une commande, que les composants nécessaires s'y trouvent. Et nous, tout ce qu'il nous faut, c'est d'y ajouter le chemin vers l'exécutable de PHP.

Pour ce faire, il suffit de:

- ajouter un point-virgule tout à la fin de la liste ;

---

1. %WAMPDIR% est à remplacer par le chemin de l'installation de WAMP

2. %APACHE\_VERSION% est à remplacer par la version d'Apache, avec les points

3. %WAMPDIR% est à remplacer par le chemin de l'installation de WAMP

4. Idem

5. Local Area network, réseau local. Typiquement, votre routeur/box Internet et les appareils qui s'y connectent (mobile, ordinateur, lecteur Blu-ray...) forment un LAN

6. Wide Area network, réseau étendu. On comprend par là tout ce qu'il y a au-delà de votre routeur/box, et par extension Internet

## II. Utiliser son serveur

— coller à sa suite le chemin vers le dossier où se trouve PHP, soit `%WAMP  
DIR%\bin\php\php%PHP_VERSION%`<sup>3, 4</sup>



N'ajoutez aucune espace avant ni après le point-virgule

[Retourner au texte.](#)

## II.2. ... ou pour des besoins spécifiques

### Introduction

Nous avons vu deux-trois choses pour pouvoir utiliser notre serveur sans trop de casse-têtes jusqu'à maintenant. Mais le but d'un serveur de développement, c'est d'être le plus proche possible d'un environnement de production.

Du coup, la version de PHP n'est pas la même, ou la version de MySQL. Faut-il installer une autre version de WAMP pour cela ?

Certains frameworks utilisent aussi PHP en ligne de commandes. Comment mettre ça en place ?

Ce chapitre est pensé pour vous qui vous posez ces questions. 🍊

### II.2.1. WAMP, ses modules, et Microsoft Visual C++

Comme expliqué lors de l'installation, WAMP n'est pas le nom de la solution installée, mais un acronyme qui désigne un environnement : Windows Apache MySQL et PHP. Du coup, il paraît facile de comprendre que WampServer possède une architecture modulaire.

Cette architecture permet d'avoir plusieurs versions des différents composants. Vous pouvez donc avoir autant de versions de PHP que nécessaire, autant de versions de MySQL, voire autant de versions d'Apache.

Ces différentes versions sont téléchargeables à divers endroits, mais s'il y a un point dont il faut tenir compte, c'est **la version compatible de Microsoft Visual C++**.

Lassé de devoir réexpliquer quelle version est nécessaire pour les diverses versions de WAMP ou des add-ons, le mainteneur actuel fait désormais installer au moins 5 versions, alors que dans la pratique, l'ordinateur n'en utilisera probablement pas plus que trois pour faire fonctionner WAMP. L'avantage étant que si vous voulez de nouveaux add-ons, vous ne devriez pas avoir à installer quoi que ce soit d'autre. 🍊

Les versions de Microsoft Visual C++ sont souvent mentionnées sur les pages de téléchargement, et parfois aussi dans le nom de l'archive du module que vous téléchargez. Seulement, comme *Microsoft Visual C++* est long à écrire, c'est souvent raccourci en *Visual C++* suivi de l'année, ou abrégé **VC**, suivi d'un numéro correspondant à une version. Pour ne pas faire simple, même si l'on parle de Microsoft Visual C++ 2012, ce ne sera pas VC2012 ou VC12, mais VC11...

Afin de vous simplifier la tâche, voici un tableau avec la correspondance entre différentes versions abrégées et leur version complète. Les versions de PHP sont mentionnées d'après les informations de la page [windows.php.net/download/](http://windows.php.net/download/) ↗

Code de version	Nom complet	Fichier signalé manquant	Téléchargement
-----------------	-------------	--------------------------	----------------

## II. Utiliser son serveur

VC17 (mise à jour rétro-compatible depuis VC14, PHP 7.0, 7.1, 7.2, 7.3, 7.4, 8.0, 8.1, WAMP 3.x)	Microsoft Visual C++ 2022	VCRUNTIME140.dll, msvc140.dll	<a href="#">x86</a> , <a href="#">x64</a>
VC14, VC15, VC16 (obsolètes, voir la première ligne du tableau)	Microsoft Visual C++ 2015, 2017, 2019		Voir la première ligne du tableau
VC12 (WAMP 3.x)	Microsoft Visual C++ 2013	msvcr120.dll (msvcr130.dll)	<a href="#">x86</a> , <a href="#">x64</a>
VC12 update 5 (WAMP 3.x)	Mise à jour pour Visual C++ 2013 et Visual C++ Redistributable Package		<a href="#">x86</a> et <a href="#">x64</a> (choisir dans la page)
VC11 (WAMP 2.5 et 3.x, PHP 5.5, 5.6)	Microsoft Visual C++ 2012 (update 4)	msvcr110.dll	<a href="#">x86</a> et <a href="#">x64</a> (choisir dans la page)
VC10 (WAMP 3.x)	Microsoft Visual C++ 2010 SP1	msvcr100.dll	<a href="#">x86</a> et <a href="#">x64</a> (choisir dans la page)
VC9 (WAMP 2.4 et 3.x, PHP 5.3, 5.4)	Microsoft Visual C++ 2008	msvcr90.dll	<a href="#">x86</a> , <a href="#">x64</a>
KB# (spécifique à votre version de Windows)	Mise à jour de sécurité du Runtime C universel	api-ms-win-crt-runtime-l1-1-0.dll	Choisir dans la page pour votre version exacte de Windows

TABLE II.2.2. – Abréviations et noms complets des versions de Microsoft Visual C++  
Ainsi, faites attention au nom de l'archive du module que vous téléchargez s'il comporte une partie VC#, et regardez aussi ce qui est expliqué sur la page d'où vous téléchargez le module.

*i*

Notez que, comme les versions de Visual C++ sont utilisées par beaucoup de programmes sous Windows, il est très possible d'avoir déjà une version compatible de Visual C++ d'installée. Néanmoins, il est conseillé de réinstaller les versions ci-dessus en veillant bien à utiliser le clic droit > *Exécuter en tant qu'administrateur*, et si proposé, de réparer les installations existantes.

*x*

Ne prenez **SURTOUT PAS** juste un fichier manquant pour le placer directement dans `C:\Windows\system32`. Vous n'avez aucune assurance que le fichier soit le bon ni qu'il ne fasse pas autre chose que ce qu'il devrait, sans compter que ce ne sera pas nécessairement le seul fichier à aller récupérer.

Toute manipulation qui vous fait toucher aux dossiers système est à considérer comme douteuse—ce n'est pas pour rien que Windows vous demande une validation quand vous y accédez.



A vos risques et périls.

## II.2.2. Plusieurs versions de PHP

L'un des atouts intéressants de WAMP est que l'on peut, avec la même installation, avoir plusieurs versions de PHP, afin de travailler avec une version qui correspond à celle qui sera utilisée en production. En effet, WAMP 3.2 est livré avec PHP 8.1, qui n'est pas forcément disponible chez tous les hébergeurs. Il serait donc intéressant de pouvoir utiliser la même version que celle qui sera utilisée en production.

### II.2.2.1. Télécharger les binaires pour Windows

Dans un premier temps, il vous faudra ce qu'on appelle les *binaires* de PHP pour la version que vous souhaitez.

Les binaires sont simplement des fichiers compilés et prêts à l'utilisation, au contraire des *sources*.

Comme dans PHP il y a plusieurs extensions, cela fait beaucoup de fichiers, et une arborescence à respecter. C'est pourquoi vous téléchargerez très probablement une archive zip.

Vous pouvez les télécharger depuis divers endroits, néanmoins nous vous conseillons deux sites.

#### II.2.2.1.1. PHP for Windows

Y accéder [↗](#)

Le site officiel, d'où vous pouvez télécharger tout un éventail de versions de PHP, bêtas comme stables. C'est le site à surveiller pour être au courant des sorties de nouvelles versions que vous pourrez utiliser avec WAMP.

Néanmoins, la page de téléchargement peut paraître déroutante : pour les versions actuellement maintenues de PHP, il n'y a pas moins de 4 versions à choix ! 🍊 Laquelle choisir ?

Parmi les versions proposées, vous verrez qu'elles portent la mention *Thread Safe* ou *Non Thread Safe* (parfois abrégées *TS* et *NTS*). Sans entrer dans les détails, ce qu'il nous faut, c'est une version **Thread Safe**, Apache gérant le multithreading.

Ensuite, choisir entre **x86** et **x64** dépend de l'installation de WAMP que vous avez faite.



Une installation **x64** de WAMP vous oblige à avoir des versions **x64** de PHP. Cela ne fonctionne pas comme Windows avec lequel vous pouvez installer des programmes **x86** qui fonctionnent alors que vous avez un Windows **x64**.



Mais j'ai vraiment besoin de vieilles versions avant 5.6, je fais comment ? 🍊

On espère que c'est pour préparer une migration de projet. 🍊

**II.2.2.1.1.1. Pour PHP 5.2 à 5.5 x86 et PHP 5.5 x64** Vous pouvez utiliser [les archives de windows.php.net](#) [↗](#). La page peut paraître austère et il semble y avoir plein de choses à télécharger, mais souvenez-vous de ce qui a été dit ci-dessus : il ne faut pas prendre de fichier avec **nts** dans le nom, et basez-vous sur le **x86** ou **x64** plutôt que sur Win32

**II.2.2.1.1.2. Pour PHP 5.2 à 5.4 x64** Vous pourrez trouver votre bonheur sur le site ci-après.

### II.2.2.1.2. Les forums d'Apache Lounge

[Y accéder](#) ↗

Ce site a l'avantage de proposer diverses versions de PHP **x86** comme **x64**, et notamment pour PHP 5.3 et 5.4, alors que ce n'est que depuis PHP 5.5 que PHP est fourni de manière officielle pour Windows avec architecture **x64**. Les désavantages sont que les nouvelles versions de PHP ne s'y trouvent pas dès leur sortie, et les différentes extensions de PHP ne sont pas nécessairement les mêmes versions que pour PHP officiel—même si ce dernier point porte peu souvent à conséquence.

A vous de trouver dans le forum les sujets qui proposent les versions qui vous intéressent, mais la plupart sont regroupées dans [ce sujet](#) ↗.

Là aussi, vous pouvez avoir plusieurs choix pour chaque version. Certaines comportent *NTS* dans leur nom et ne sont donc pas à prendre.



Ne vous laissez pas dérouter par les éventuelles mentions Win32 et Win64 dans le nom des fichiers, ce n'est pas l'indication qu'il s'agit d'une version Windows 32 ou 64 bits. Seules les mentions **x86** et **x64** sont à prendre en compte à ce propos.

### II.2.2.2. Installer la nouvelle version de PHP

Une fois l'archive téléchargée, il faut l'extraire dans un dossier particulier.

Si vous vous êtes déjà un peu promené dans le dossier d'installation de WAMP, vous aurez remarqué que les fichiers relatifs à PHP (ainsi qu'à MySQL ou à Apache) sont dans des chemins de ce genre : `%WAMPDIR%\bin\php\php5.5.121`.

Imaginons maintenant que vous ayez téléchargé PHP 5.5.38. Il vous faudra donc extraire l'archive dans le dossier `%WAMPDIR%\bin\php\php5.5.382`.

Une fois l'archive extraite, vous devriez donc avoir les dossiers suivants (entre autres) :

- `%WAMPDIR%\bin\php\php5.5.38\dev3`
- `%WAMPDIR%\bin\php\php5.5.38\ext4`
- `%WAMPDIR%\bin\php\php5.5.38\extras5`

Maintenant, il ne suffit pas d'avoir extrait l'archive au bon endroit : WAMP a besoin de deux autres fichiers pour pouvoir fonctionner avec cette nouvelle version de PHP.

#### II.2.2.2.1. Le fichier `wampserver.conf`

Ce fichier contient un bref script PHP qui permet à WAMP de configurer correctement la liaison avec Apache. En règle générale, vous pouvez le copier depuis une version de PHP (les deux premiers numéros doivent néanmoins concorder) qui fonctionne déjà avec WAMP et la coller dans le dossier où vous avez extrait l'archive, soit dans `%WAMPDIR%\bin\php\php5.5.386`.

```
1 <?php
2
3 $phpConf['phpIniDir'] = '.';
4 $phpConf['phpExeDir'] = '.';
```



```
5 $phpConf['phpConfFile'] = 'php.ini';
6
7 $phpConf['apache']['2.2']['LoadModuleName'] = 'php5_module';
8 $phpConf['apache']['2.2']['LoadModuleFile'] = 'php5apache2_2.dll';
9 $phpConf['apache']['2.2']['AddModule'] = '';
10
11 $phpConf['apache']['2.4']['LoadModuleName'] = 'php5_module';
12 $phpConf['apache']['2.4']['LoadModuleFile'] = 'php5apache2_4.dll';
13 $phpConf['apache']['2.4']['AddModule'] = '';
14
15 ?>
```

Listing 19 – Exemple de fichier `wampserver.conf` pour PHP 5

Les lignes 3 à 5 sont les indications pour trouver les exécutables de PHP et le fichier principal de configuration pour la ligne de commandes. Les lignes 7 à 8 pour Apache 2.2, comme le dénote le second index '2.2', et les lignes 11 à 13 pour Apache 2.4.

Avec PHP 8, les lignes 7 à 9 disparaissent, parce que PHP 8 n'est pas compatible avec Apache 2.2.

Pour certaines versions de PHP, ou selon la source, il y a quelque chose de plus à modifier. Certains binaires de PHP ont été fournis avec un fichier `php%MAJOR%.dll`<sup>7</sup> qui est l'équivalent renommé de `php%MAJOR%apache2_4.dll`<sup>8</sup>. Deux options s'offrent à vous :

- renommer le fichier `php%MAJOR%.dll`<sup>9</sup> en `php%MAJOR%apache2_4.dll`<sup>10</sup> et copier-coller le fichier `wampserver.conf` depuis une autre version de PHP;
- changer les lignes du fichier `wampserver.conf` repris depuis une autre version de PHP, afin de remplacer cette fois `php%MAJOR%apache2_4.dll`<sup>11</sup> par `php%MAJOR%.dll`<sup>12</sup>.

Dans les deux cas, les deux premiers numéros de la nouvelle version et de celle dont vous reprenez le fichier doivent concorder.



### PHP 8

Pour PHP 8, c'est `php8_module` qu'il faut remplacer par `php_module`, le numéro de version ayant disparu du nom du module pour Apache dès cette version. De plus, les lignes contenant '2.2' peuvent toutes être entièrement enlevées du fichier, PHP 8 n'est plus compatible avec Apache 2.2.

#### II.2.2.2. Les fichiers `php.ini`

WAMP a pour particularité d'utiliser deux ou trois fichiers `php.ini`

1. `%WAMPDIR%\bin\php\php5.5.38\php.ini`<sup>13</sup> : c'est celui qui sera pris en compte par PHP en ligne de commandes. Vous pouvez vérifier son emplacement avec la commande `php --ini`.
2. `%WAMPDIR%\bin\php\php5.5.38\phpForApache.ini`<sup>14</sup> : c'est la sauvegarde de la configuration de PHP pour Apache.
3. `%WAMPDIR%\bin\apache\apache2.4.9\bin\php.ini`<sup>15</sup>  
Depuis WAMP 2.5 et plus, c'est un *lien symbolique* vers le N° 2. Auparavant, c'était une *copie* du N° 2. Ce dernier fichier était créé lors du changement de version de PHP ou au premier démarrage. C'était donc la version qui faisait foi pour le module Apache, celui

## II. Utiliser son serveur

qui était utilisé le plus souvent quand on accédait par HTTP au serveur, et *de facto* celle qui était mentionnée dans `phpinfo`.

*i*

Il est intéressant de vérifier quel fichier est *réellement* pris en compte dans la ligne de commandes et dans `phpinfo`, surtout si vous êtes passé d'une autre solution à WAMP, ou que vous l'avez mis à jour, et que vous avez des problèmes.

Vous n'avez pas à aller dupliquer votre fichier `php.ini` dans le répertoire d'Apache, mais il vous faut cependant créer `phpForApache.ini`.

*i*

Certaines distributions de PHP sont fournies avec plusieurs fichiers de configuration, notamment parfois `php.ini` et `php-cli.ini`, voire aussi `php-apache.ini`. PHP sait de lui-même que :

- `php-cli.ini` est pour la ligne de commandes
- `php-apache.ini` est pour Apache

Mais il faut cependant copier `php-apache.ini` et renommer la copie en `phpForApache.ini` pour que WAMP fonctionne correctement.

### II.2.2.3. Eviter les incompatibilités avec ICU sous WAMP 2.5 (de 2.5.0 à 2.5.4 compris)

Plus le temps passe, et plus les versions de PHP comme de ses extensions s'améliorent. Mais cela implique aussi que certaines versions des extensions finissent par être considérées comme incompatibles avec les versions de PHP, et vice-versa.

D'autre part, certaines extensions de PHP ne sont pas uniquement un fichier DLL placé dans le répertoire *ad-hoc*, mais dépendent elle-mêmes de bibliothèques externes qui évoluent aussi. Par exemple, il y a l'extension `intl`, qui se base sur la bibliothèque ICU.

Sous Windows, cette dernière se présente sous une série de fichiers DLL dont le nom commence par `icu`, suivi de plusieurs caractères, et d'un numéro de version. Ces fichiers se trouvent dans le même dossier que le fichier `php.ini`, mais c'est Apache qui en a besoin.

Seulement, Windows n'ira chercher les fichiers là où ils se trouvent uniquement si la version de PHP avec laquelle ils ont été fournis est celle que vous utilisez en ligne de commandes. Or, quand vous avez plusieurs versions de PHP, il n'est pas garanti que la bibliothèque ICU pour la version de PHP en ligne de commandes soit compatible avec la version de l'extension `intl`..

Les concepteurs de WAMP ont pensé à cela, et ont prévu de quoi gérer les différentes versions d'ICU.

Pour éviter aux utilisateurs de WAMP d'avoir à se soucier de fichiers qu'il faudrait déplacer pour faire fonctionner leur serveur ainsi que pour éviter de doubler des fichiers, WAMP crée des raccourcis vers ceux-ci dans le dossier d'Apache. La liste des fichiers à traiter ainsi est placée dans le code de `%WAMPDIR%\scripts\config.inc.php`<sup>16</sup>, et se présente initialement comme suit :

```
62 <?php
63 $phpDllToCopy = array (
```

## II. Utiliser son serveur

```
64     'icudt52.dll', //[modif oto] - Ajouts pour éviter unknown
        error PHP 5.6.a2
65     'icuin52.dll',
66     'icui52.dll',
67     'icule52.dll',
68     'iculx52.dll',
69     'icutest52.dll',
70     'icutu52.dll',
71     'icuuc52.dll',
72     'icudt51.dll', //[modif oto] - Ajouts pour éviter unknown
        error PHP 5.5.6
73     'icuin51.dll',
74     'icui51.dll',
75     'icule51.dll',
76     'iculx51.dll',
77     'icutest51.dll',
78     'icutu51.dll',
79     'icuuc51.dll',
80     // ...
```

Listing 20 – Extrait de %WAMPDIR%\scripts\config.inc.php<sup>17</sup> fourni avec WAMP 2.5.0  
Comme vous pouvez le constater, il s'agit vraiment de la liste des fichiers. Il est possible de permettre de les gérer avec plus de facilité. Plutôt que d'avoir à ajouter 8 lignes à chaque nouvelle version d'ICU, vous pouvez remplacer les lignes 63 à 111 par les suivantes, plus aisément maintenables et extensibles :

```
62 <?php
63 $icu = array(
64     'number' => array(
65         '54', // PHP 5.6.4
66         '53', // PHP 5.6
67         '52', // PHP 5.6a2
68         '51', // PHP 5.5.6
69         //'50', // PHP 5.5
70         //'49', // PHP 5.3 / 5.4
71     ),
72     'name' => array(
73         'icudt',
74         'icuin',
75         'icui5',
76         'icule',
77         'iculx',
78         'icutest',
79         'icutu',
80         'icuuc',
81     ),
82 );
83 $php_icu_dll = array();
```

```
84 foreach($icu['number'] as $icu_number) {
85     foreach($icu['name'] as $icu_name) {
86         $php_icu_dll[] = $icu_name . $icu_number . '.dll';
87     }
88 }
89
90 $phpDllToCopy = array_merge(
91     $php_icu_dll,
92     array (
93         'libeay32.dll',
94         'libsasl.dll',
95         'libssh2.dll', // PHP 5.5.17
96         'php5isapi.dll',
97         'php5nsapi.dll',
98         'ssleay32.dll',
99         'php5ts.dll',
100     ),
101 );
```

Listing 21 – Remplacement pour la gestion des versions de la bibliothèque ICU. Les lignes en surbrillance sont les ajouts de versions d'ICU pour les versions plus récentes de PHP que la 5.5.12 fournie avec WAMP 2.5.0

Ainsi, dès qu'une nouvelle version d'ICU est fournie avec PHP, vous n'avez qu'à ajouter le numéro de version dans la liste, entre les lignes 64 et 65. Si de plus un nouveau fichier est ajouté, alors vous pouvez mettre son nom (**sans la partie du numéro de version**) à la suite de la ligne 80.

Voilà, il vous suffit maintenant d'apporter les modifications souhaitées aux fichiers de configuration de PHP, puis de redémarrer WAMP.

Si vous cliquez sur l'icône de WAMP dans la barre de statut et que vous allez sur *PHP > version*, vous devriez désormais avoir 5.5.58 en plus du reste dans la liste. Choisissez cette version. WAMP va redémarrer de lui-même. Rendez-vous à l'adresse <http://localhost/?phpinfo> pour vérifier que tout va bien, et que c'est bien PHP 5.5.38 qui fonctionne désormais.

Pour repasser à une autre version de PHP, même procédé : clic sur l'icône de WAMP dans la barre de statut > *PHP > Version* et choisissez-en une autre.

Vous avez maintenant plusieurs versions de PHP avec une seule installation de WAMP ! 🎉

### II.2.2.4. Mettre à jour vers une révision plus récente



Moi j'aimerais juste avoir la dernière version de PHP, je n'ai plus besoin des autres, je peux les supprimer ?

Oui, mais il faut commencer par installer cette nouvelle version comme expliqué ci-dessus et passer à cette version.

Le noyau de WAMP se base sur PHP pour fonctionner en interne. Il lui faut donc savoir quelle version il doit utiliser ainsi que où la trouver, et la seule qu'on puisse être sûr qu'elle soit présente dès l'installation et donc dont on sait où elle se trouve, c'est celle qui est fournie avec. 🍊

Mais comme tout bon programmeur, l'équipe qui travaille sur WAMP a prévu un et un seul

## II. Utiliser son serveur

endroit où on peut changer très simplement la version à utiliser.

Depuis WampServer 3.1.1, il existe un menu pour ce faire : clic droit sur l'icône de WampServer > *Outils* > *Changer version PHP CLI*.

Avec une version moins récente, il faut effectuer la modification ci-dessous **alors que WampServer est totalement arrêté**.

Rendez-vous dans le dossier d'installation de WAMP, et ouvrez le fichier `wampmanager.conf` avec un éditeur de texte. Malgré l'extension, c'est le même format de fichier INI que celui utilisé pour configurer PHP et MySQL.

Recherchez alors les lignes suivantes :

```
1 [php]
2 phpCliVersion = "5.6.40"
```

Listing 22 – Bref extrait du contenu du fichier `wampmanager.conf`

Vous l'aurez compris, il suffit de changer le vieux numéro de version par le nouveau. 🧙

Vous n'avez pas trop besoin de vous soucier d'éventuels problèmes d'incompatibilité, les scripts de WAMP sont très basiques, et n'utilisent pas de fonctions très avancées.



Si vous aviez prévu une version de PHP en ligne de commandes, assurez-vous qu'elle n'est pas parmi celles que vous allez supprimer.

### II.2.3. Plusieurs versions du SGBD



WAMP, depuis la version 3.2.0, installe MariaDB par défaut, une alternative *open source* à et compatible avec MySQL, détenu par Oracle. **Il n'est pas nécessaire de passer de MariaDB à MySQL.**

Vous vous en doutez, s'il y a moyen d'avoir plusieurs versions de PHP avec WAMP, il y a aussi moyen d'avoir plusieurs versions de SGBD. 🍏

1. %WAMPDIR% est à remplacer par le dossier d'installation de WAMP
2. Idem
3. Idem
4. Idem
5. Idem
6. Idem
7. %MAJOR% est à remplacer par le numéro de version majeure de PHP installée (5, 7, 8)
8. Idem
9. Idem
10. Idem
11. Idem
12. Idem
13. %WAMPDIR% est à remplacer par le dossier d'installation de WAMP
14. Idem
15. Idem
16. Idem
17. Idem
18. Idem

## II. Utiliser son serveur

Cependant, c'est un peu plus difficile à mettre en place, surtout si vous souhaitez utiliser la version fournie avec WAMP 2.5 (MySQL 5.6.17) avec une version 5.5 et partager les bases de données. Nous allons voir cela en temps utile.

### II.2.3.1. Installer la nouvelle version

Pour trouver la version qui vous plaît, rien de mieux que [le site officiel de MariaDB](#) ou [celui de MySQL](#). Naviguez dans le site afin de trouver la version qu'il vous faut.



Ne prenez **PAS** les *MSI Installer*, qui sont mis en avant, mais prévus pour des installations séparées des composants d'un serveur.

Prenez l'**archive ZIP** correspondant à votre installation de WAMP, **x86** ou **x64**.



Le site de MySQL est réfléchi pour que vous pensiez avoir à vous enregistrer afin de télécharger quelque chose, mais il n'en est rien: en bas de la page où les options d'authentification et d'enregistrement s'affichent, il y a le texte *No thanks, just start my download*. Cliquez dessus pour que commence votre téléchargement.

Une fois cette archive téléchargée, extrayez-là simplement dans le répertoire `%WAMPDIR%\bin\%SGBD_NAME%\%SGBD_NAME%%VERSION%`<sup>1</sup>, en remplaçant `%VERSION%` par le numéro de version (avec les points) de MySQL que vous voulez installer.

On va encore s'occuper de la configuration de la version de MySQL. Comme point de départ, on va utiliser un fichier de configuration `%WAMPDIR%\bin\%SGBD_NAME%\%SGBD_NAME%%VERSION%\my.ini`<sup>2</sup> (en remplaçant `%VERSION%` par quelque chose de précédemment installé), le copier-coller dans le dossier de la nouvelle version, et ouvrir cette copie.

Remplacez toutes les occurrences de l'ancienne version par la version que vous venez d'installer, et éventuellement les occurrences du nom du **SGBD** afin de renseigner correctement les chemins.

#### II.2.3.1.1. Pour une version 5.5 ou moins récente

Dans le cas où vous installez une version 5.5 ou moins récente de MySQL, il y a une autre modification à effectuer.

Localisez les deux lignes qui sont comme celles ci-dessous et commentez-les en ajoutant un `;` devant :

```
1 explicit_defaults_for_timestamp = TRUE
```

Listing 23 – Lignes qui paramètrent un comportement lié aux colonnes de type `timestamp`. Finalement, comme on l'a vu pour PHP, il faut mettre un fichier `wampserver.conf` dans le dossier où l'on vient d'extraire la version souhaitée de MySQL, afin que WAMP puisse savoir comment utiliser cette nouvelle version. Nous n'allons pas faire dans la dentelle parce que ce n'est pas nécessaire: récupérez le fichier `wampserver.conf` dans le dossier `%WAMPDIR%\bin\%SGBD_NAME%\%SGBD_NAME%%VERSION%`<sup>3</sup> (remplacez `%VERSION%` par celui d'une version déjà fonctionnelle) et copiez-collez-le dans le dossier de la nouvelle version.

Si vous redémarrez WAMP, que vous cliquez sur l'icône dans la barre de statut et que vous allez regarder dans MySQL > Version, vous verrez apparaître, en plus de la 5.6.17, celle que vous

## II. Utiliser son serveur

venez d'installer.

### II.2.3.2. Partager les bases de données entre plusieurs versions

Il est légitime de vouloir retrouver ses données même si on change de version de SGBD. Seulement pour cela, il n'y a pas d'autre manière sûre de faire un export des données depuis l'ancienne vers la nouvelle version.

#### II.2.3.2.1. Sauvegarder les données

Pour ce faire, vous trouverez les informations dans [la partie du tutoriel expliquant les sauvegardes](#) [↗](#). Faites les manipulations, puis fermez totalement WAMP.

#### II.2.3.2.2. Préparer phpMyAdmin pour la nouvelle version de MySQL

Notre nouvelle version de MySQL n'est pas totalement configurée, notamment au niveau des droits d'accès. Or, phpMyAdmin se base dessus: si vous avez un mot de passe pour l'utilisateur root de la précédente version de MySQL, ce mot de passe ne sera pas reconnu, et vous ne pourrez donc pas accéder à phpMyAdmin pour la nouvelle version de MySQL que l'on vient d'installer. Il faut remettre la configuration par défaut, soit root comme utilisateur et aucun mot de passe.

La configuration de phpMyAdmin se fait dans le fichier `%WAMPDIR%\apps\phpMyAdmin%VERSION%\config.inc.php`<sup>4</sup>. Nous allons en conserver une copie en l'état: copiez-collez-le dans le même dossier. Vous aurez ainsi un fichier `config.inc.php` et un autre `config.inc - Copie.php`.

Ouvrez `config.inc.php` avec votre éditeur de code favori, et remplacez le contenu par ce qui suit.

```
1 <?php
2 /* Servers configuration */
3 $i = 0;
4
5 $cfg['blowfish_secret'] = 'éhP)(KSds';
6
7 /* Server: localhost [1] */
8 $i++;
9 $cfg['Servers'][$i]['host'] = '127.0.0.1';
10 // Laissez "mysqli" si vous l'avez d'activé sur votre serveur
11 // Si non, mettez simplement "mysql"
12 $cfg['Servers'][$i]['extension'] = 'mysqli';
13 $cfg['Servers'][$i]['auth_type'] = 'config';
14 $cfg['Servers'][$i]['user'] = 'root';
15 $cfg['Servers'][$i]['password'] = '';
16 // Autorise connexion sans mot de passe
17 $cfg['Servers'][$i]['AllowNoPassword'] = true;
18
19 // Supprime l'avertissement aux sujets des tables pmadb
20 $cfg['PmaNoRelation_DisableWarning'] = true;
21
```

```
22 /* End of servers configuration */
```

Listing 24 – Contenu qui remplace temporairement ce qu’il y avait dans `%WAMPDIR%\apps\phpMyAdmin%VERSION%\config.inc.php`<sup>5</sup> pour préparer le ”partage de bases de données”

### II.2.3.2.3. Initialiser la nouvelle version de MySQL

Maintenant, il faut initialiser la nouvelle version. Lancez WAMP comme d’habitude. Puis, quand l’icône est verte, cliquez dessus et rendez-vous dans `MySQL > Version` et activez la nouvelle que nous venons d’installer. L’icône va virer au rouge, puis devrait revenir au vert. Si c’est le cas, vous pouvez continuer. Si l’icône reste orange, fermez WAMP et revérifiez les différentes modifications apportées au fichier `my.ini` plus haut.

### II.2.3.2.4. Importer les données (et surtout les utilisateurs)

Pendant la prochaine étape, il nous faut renseigner la nouvelle version sur les utilisateurs que nous avons déjà. Pour ce faire, récupérez le contenu du fichier de sauvegarde contenant les utilisateurs que vous avez créé [plus haut](#) [↗](#) en l’ouvrant et le copiant.

Rendez-vous maintenant dans phpMyAdmin, onglet SQL. Collez-y le contenu du fichier de sauvegarde des utilisateurs et exécutez la suite de requêtes, puis quittez phpMyAdmin. Il ne devrait plus être accessible désormais, on va s’en charger.

### II.2.3.2.5. Remettre la configuration d’origine de phpMyAdmin

Maintenant qu’on a renseigné la nouvelle version de MySQL sur les utilisateurs, on peut remettre la configuration de phpMyAdmin qui en tenait compte.

Retournez dans le dossier `%WAMPDIR%\apps\phpMyAdmin%VERSION%`<sup>6</sup>, et :

1. renommez le fichier `config.inc.php` en `config-change_version.inc.php`, afin de l’avoir encore sous la main dans le cas où vous souhaiteriez installer une autre version de MySQL.
2. faites encore une copie du fichier `config.inc - Copie.php` qui va s’appeler `config.inc - Copie (2).php`...
3. ... et renommez ce dernier en `config.inc.php`.

Vous avez donc remis la configuration d’origine pour phpMyAdmin, mais vous utilisez l’autre version de MySQL. Le fichier `config.inc - Copie.php` est une sauvegarde qui devrait fonctionner si vous revenez à la précédente version de MySQL.

Voilà, on en a terminé pour la cohabitation de deux (ou plus) versions de vos **SGBD** avec la même installation de WAMP. Il ne nous reste pas grand-chose à voir pour que vous puissiez travailler dans les meilleures conditions. 🍊

---

1. `%WAMPDIR%` est à remplacer par votre dossier d’installation de WAMP, `%SGBD_NAME%` par le nom en minuscules de votre **SGBD**

2. Idem

3. Idem

4. `%WAMPDIR%` est à remplacer par votre dossier d’installation de WAMP, `%VERSION%` est à remplacer par le numéro de version de phpMyAdmin, avec les points de séparation

5. Idem

6. Idem



## II.2.4. Générer des certificats SSL/TLS et activer HTTPS

Auparavant, le protocole HTTPS était utilisé principalement pour opérations sensibles effectuées en ligne. Si c'était autant restreint, c'était parce que les connexions sécurisées représentent tout de même une utilisation supplémentaires de ressources tant client que serveur. Mais avec les problèmes de vie privée sur Internet, de nombreux sites ont opté pour cette possibilité sur l'entier de leurs pages, notamment les réseaux sociaux. Et les machines devenant plus performantes, la charge supplémentaire impliquée est minimisée, au point que le protocole historique en clair disparaîtra probablement à moyen terme.

Evidemment, en tant que développeur, cela implique de pouvoir utiliser ce protocole sur la version de travail de nos sites. D'ailleurs, si vous utilisez le nom de domaine de premier niveau `.dev`, Chrome et Firefox notamment forcent le protocole HTTPS. Or, WAMP n'est pas paramétré pour ça par défaut. Mais comme c'est un serveur Apache comme une grande partie des serveurs actuellement utilisés dans le monde, il est possible de configurer tout ça. La manière présentée ci-après est une version fonctionnelle simplifiée.

i

L'utilisation d'un certificat SSL demande un nom de domaine, donc *a minima* un hôte virtuel auquel lier le certificat

### II.2.4.1. Générer un certificat auto-signé

Une des possibilités les plus simples est de créer un certificat auto-signé pour l'hôte virtuel qui en a besoin. Ceci peut se faire aisément avec divers outils, mais nous allons ici nous servir d'un élément qui est fourni avec WAMP: openssl. Comme son nom l'indique aux amateurs, il s'agit de l'implémentation libre et gratuite des outils pour gérer le protocole SSL.

#### II.2.4.1.1. Une ligne de commandes

La génération du certificat va s'effectuer avec une seule commande, qui demandera quelques informations lors de son exécution.

```
1 %WAMPDIR%\bin\apache\apache%APACHE_VERSION%\bin\openssl.exe req
  --config=%WAMPDIR%\bin\apache\apache%APACHE_VERSION%\conf\open_
  ssl.conf -x509 -out %DOMAIN_NAME%.crt -keyout
  %DOMAIN_NAME%.key -newkey rsa:2048 -nodes -sha256
```

Listing 25 – Un aperçu de la commande. `%WAMPDIR%` est à remplacer par le chemin d'installation de WAMP ; `%APACHE_VERSION%` par la version d'Apache, **avec les points de séparation**; `%DOMAIN_NAME%` par le nom de domaine que vous souhaitez certifier. Plutôt que d'enregistrer les fichiers que cette commande va générer dans le dossier de configuration d'Apache, il serait plus intéressant d'utiliser un dossier en-dehors de celui d'Apache afin de faciliter les mises à jour le cas échéant, et de les conserver en cas de désinstallation de WampServer pour passer à une autre solution. Un dossier `ssl`, au même niveau que `www`, serait plus pratique. Créez le dossier et rendez-y vous avec la ligne de commandes avant de lancer ce que vous aurez adapté.

### II.2.4.1.2. Votre certificat pour localhost

Histoire de se simplifier la tâche, on va créer pour premier certificat pour l'hôte virtuel `localhost`. Cela nous permettra de ne pas avoir trop de modifications à apporter à la configuration lors de la création des certificats pour d'autres hôtes virtuels. Adaptez donc la commande en remplaçant `%DOMAIN_NAME%` par `localhost`. La sortie ressemblera à ce qui suit:

```
1 %WAMPDIR%\ssl>C:\wamp\bin\apache\apache%APACHE_VERSION%\bin\openssl
  l.exe req
  --config=C:\wamp\bin\apache\apache%APACHE_VERSION%\conf\openssl
  l.conf -x509 -out localhost.crt -keyout localhost.key -newkey
  rsa:2048 -nodes -sha256
2 Generating a RSA private key
3 .....+.....+..+++++.
  .....++++
4 .....++++
5 writing new private key to 'localhost.key'
6 -----
7 You are about to be asked to enter information that will be
  incorporated
8 into your certificate request.
9 What you are about to enter is what is called a Distinguished Name
  or a DN.
10 There are quite a few fields but you can leave some blank
11 For some fields there will be a default value,
12 If you enter '.', the field will be left blank.
13 -----
14 Country Name (2 letter code) [AU]:FR
15 State or Province Name (full name) []:Île de France
16 Locality Name (eg, city) []:Paris
17 Organization Name (eg, company) []:Zeste de Savoir
18 Organizational Unit Name (eg, section) []:WAMP : développer avec
  PHP / AJAX / HTML sous Windows
19 Common Name (e.g. server FQDN or YOUR name) []:localhost
20 Email Address []:
```

Listing 26 – Exemple d'exécution de la commande avec OpenSSL 1.1.1g

Aux lignes 14 à 20, la commande demande une série d'informations qui peuvent varier selon votre installation. Il peut y avoir des lignes en moins ou des valeurs par défaut entre les crochets avant les deux-points (comme le `[AU]` de la ligne 14), mais le plus important se trouve à la ligne 19 : c'est là que vous renseignez le nom de domaine ciblé. Notez cependant :

- que selon votre version de WAMP lors de l'installation, toutes ces informations ne seront pas nécessairement demandées ;
- que les valeurs autres que le « *Common Name* » ne sont pas obligatoires, même s'il est recommandé de les remplir : il n'est pas impossible que votre navigateur se méfie plus des certificats n'ayant pas ces informations à terme ;
- que la ligne 12 explique qu'il faut saisir un unique point pour n'avoir absolument aucune valeur. En réalité, c'est un peu plus subtil et n'a d'utilité dans l'exemple ci-dessus que pour la ligne 14. Si l'on avait simplement appuyé sur `ENTER` en pensant laisser l'option

## II. Utiliser son serveur

«*Country Name*» vide, c'est «AU» qui aurait figuré dans notre certificat. Pour la ligne 20 en revanche, cela n'aurait pas changé grand chose, parce que la valeur par défaut est déjà vide.

Mais sinon, c'est fait : vous avez généré deux fichiers nommés `localhost.key`, et un autre `localhost.crt` dont l'extension est reconnue par Windows, ce qui fait que le type de fichier est similaire à «Certificat de sécurité». Afin d'utiliser les certificats SSL avec notre installation de WAMP, il nous faut maintenant activer quelques modules et extensions.

### II.2.4.1.3. Les modules nécessaires et leurs réglages

Si vous ne l'avez pas déjà fait en suivant le tutoriel, il faut dans un premier temps activer l'extension PHP `php_openssl`.

Ensuite de quoi, il faut activer deux modules Apache :

1. `ssl_module`—le nom devrait être suffisamment explicite 🍊 ;
2. `socache_shmcb_module`, qui est en bref un module de cache utilisé par Apache pour le SSL.

Maintenant, il faut paramétrer ces modules.



Ne redémarrez pas Apache avant d'avoir terminé les réglages. S'il manque certaines étapes, Apache ne redémarrera pas.

Dans un premier temps, il nous faut inclure dans la configuration d'Apache les réglages pour le SSL. Rendez-vous dans le fichier `%WAMPDIR%\bin\apache\apache%APACHE_VERSION%\conf\httpd.conf`<sup>1, 2</sup> et décommentez la ligne `Include conf/extra/httpd-ssl.conf`. Vérifiez aussi que les lignes suivantes sont décommentées :

```
1 <IfModule ssl_module>
2 SSLRandomSeed startup builtin
3 SSLRandomSeed connect builtin
4 </IfModule>
```

Listing 27 – Quatre lignes à dé-commenter si besoin

Ensuite, ouvrez le fichier que nous venons d'inclure, `%WAMPDIR%\bin\apache\apache%APACHE_VERSION%\conf\extra\httpd-ssl.conf`<sup>3, 4</sup>. Il y a plusieurs choses à adapter, nous allons les traiter de haut en bas.

Première chose à vérifier : si vous avez des `${SRVROOT}`, vous pouvez les remplacer par `${INSTALL_DIR}`. Ensuite, vous trouverez la ligne `DocumentRoot "${INSTALL_DIR}/htdocs"`. Celle-ci ne correspond pas à notre installation de WAMP, il faut donc remplacer `htdocs` par `www`. On s'attend bien à ce que le contenu proposé ne change pas complètement quand on utilise `http://` ou `https://`. 🍊

Ensuite, nous allons lier les fichiers que nous avons créés à notre hôte virtuel. Localisez les deux lignes suivantes. Elles ne sont pas directement l'une sous l'autre, c'est ainsi montré ici pour faciliter les explications.

## II. Utiliser son serveur

```
1 SSLCertificateFile "${INSTALL_DIR}/conf/server.crt"
2 SSLCertificateKeyFile "${INSTALL_DIR}/conf/server.key"
```

Vous l'aurez probablement compris, les deux directives `SSLCertificateFile` et `SSLCertificateKeyFile` prennent en argument le chemin vers l'un des deux fichiers. Dans le cas qui nous occupe, il suffira de remplacer la partie `conf/server` par `ssl/localhost`—sans modifier l'extension qui correspond déjà à ce qu'il faut mettre et ce qu'on a créé.



Maintenant vous pouvez redémarrer Apache

Accédez maintenant à `https://localhost/`. Comme c'est un certificat auto-signé qui a été paramétré, votre navigateur va vous informer du risque. Vous pouvez évidemment sans autre continuer vers le site et ajouter une exception.

Vous avez créé votre premier certificat SSL, et vous avez un hôte virtuel avec accès sécurisé.



### II.2.4.1.4. Définir d'autres hôtes virtuels

Maintenant, c'est bien joli d'avoir sécurisé l'hôte virtuel `localhost`, mais cela ne sécurise pas tous les hôtes virtuels de votre WAMP en une fois, le certificat étant lié à un nom de domaine spécifique. Il faut donc re-lancer la commande pour un autre nom de domaine et créer l'hôte virtuel nécessaire pour lier le nouveau certificat. Pour simplifier, votre hôte virtuel sécurisé peut reprendre quasiment toutes les informations de l'hôte virtuel qui existe déjà et est accessible par HTTP, il y a évidemment quelques modifications et ajouts à apporter. Reprenons la définition de l'hôte virtuel qu'on avait défini dans la section sur les `<VirtualHost>` [↗](#), et effectuons les adaptations.

```
1 <IfModule ssl_module>
2 <VirtualHost *:443>
3     ServerAdmin votreemail@exemple.com
4     DocumentRoot "C:/wamp/www/monPremierSite"
5     ServerName premier-site.localhost
6     ServerAlias www.premier-site.localhost
7
8     SSLEngine on
9     SSLCertificateFile
10        "${INSTALL_DIR}/ssl/premier-site.localhost.crt"
11     SSLCertificateKeyFile
12        "${INSTALL_DIR}/ssl/premier-site.localhost.key"
13     <FilesMatch "\.(cgi|shtml|phtml|php)$">
14         SSLOptions +StdEnvVars
15     </FilesMatch>
16
17     ErrorLog "logs/premier-site.localhost-error.log"
18     CustomLog "logs/premier-site.localhost-access.log" common
19 </VirtualHost>
20 </IfModule>
```

## II. Utiliser son serveur

Listing 28 – Déclaration d'hôte virtuel pour `premier-site.localhost` adapté pour HTTPS  
Les lignes 1 et 18 sont des sécurités en cas de mise à jour d'Apache. Les modules nécessaires ne sont pas forcément activés, et `httpd-ssl.conf` pas obligatoirement chargé par la nouvelle version, on évite ainsi qu'un hôte virtuel ne pose problème parce qu'il utilise des modules indisponibles.

On a changé le port d'écoute pour l'hôte virtuel : au lieu du port HTTP 80, on a mis le port HTTPS habituel 443. A noter que le fait d'écouter ce port est déterminé dans `httpd-ssl.conf`, fichier qui a été modifié auparavant.

Les lignes 8 à 19 contiennent le minimum requis pour permettre au serveur d'utiliser l'hôte virtuel correctement : activation du moteur SSL, définition des certificat et clé, et ajout des variables d'environnement propres à l'utilisation du moteur SSL.

Maintenant, peut-être que vous vous demandez où mettre cet hôte virtuel ? Hé ben vous pouvez le mettre dans `httpd-vhosts.conf`, comme ceux qui étaient déjà prévus pour HTTP.

?

Pourquoi n'a-t-on pas mis là l'hôte virtuel pour `https://localhost/` ? Et pourquoi ne met-on pas notre nouvel hôte virtuel au même endroit que celui pour `https://localhost/` ?

L'hôte virtuel pour `https://localhost/` est déjà défini dans `httpd-ssl.conf`. Souvenez-vous, un certificat SSL est à lier avec un hôte virtuel, et on a tout fait pour `localhost` lors de l'exemple, c'est donc bien qu'on en avait un.

En revanche, il vaut mieux ne pas mettre les hôtes virtuels dans le `httpd-ssl.conf` dans la mesure où celui-ci est prévu pour la configuration du module SSL en lui-même, ce dont ne font pas partie les hôtes virtuels. `localhost` étant un hôte virtuel par défaut, il est bien lié à la configuration de SSL. D'autre part, si vous veniez à supprimer l'un ou l'autre nom de domaine, il faudrait le répercuter sur les deux hôtes virtuels. Autant qu'ils soient au même endroit pour éviter d'oublier l'un ou l'autre.

i

Mieux vaut mettre les hôtes virtuels pour le même domaine dans le même fichier, et l'un à la suite de l'autre


## Conclusion

Au final, WAMP s'avère donc être une solution souple et néanmoins sécuritaire. La seule chose qu'il n'est pas possible de faire désormais, c'est d'avoir deux versions de PHP différentes pour deux sites qui fonctionnent en parallèle. Mais si on y réfléchit, il y a une solution : installer une version x86 et une version 64 de WAMP ! Si les hôtes virtuels sont bien paramétrés, vous pourrez avoir deux serveurs simultanément sur votre machine !

1. `%WAMPDIR%` est à remplacer par le chemin d'installation de WAMP
2. `%APACHE_VERSION%` est à remplacer par la version d'Apache, avec les points de séparation
3. `%WAMPDIR%` est à remplacer par le chemin d'installation de WAMP
4. `%APACHE_VERSION%` est à remplacer par la version d'Apache, avec les points de séparation

# Conclusion

## *Conclusion*

Voilà, nous pensons que vous avez désormais de quoi vous frotter à PHP sans trop de douleur. Prenez le temps de voir les autres tutoriels parlant de PHP, regardez aussi les [FAQ](#) des différents sujets – il se peut que vous y trouviez des astuces qui vous donnent des idées 🍊 –, regardez sur le forum si quelqu'un n'a pas déjà fait ce que vous voulez faire, sans encore savoir comment ! Nous souhaitons encore remercier [Arius](#)  au passage pour le patient travail de relecture et validation effectué lors de la première publication, et à tous les autres validateurs qui sont passés par ici à chaque mise à jour. Et maintenant, bon développement !

# Liste des abréviations

- AJAX** Asynchronous Javascript and XML. 1, 9
- CSS** Cascading StyleSheet. 7
- DNS** Domain Name Server. 44, 48
- FAQ** Foire Aux Questions. 69
- HTML** HyperText Markup Language. 7, 8
- HTTP** HyperText Transfert Protocol. 8, 9
- IIS** Internet Information Service. 10, 11
- LAN** Local Area Network. 44, 48, 49
- SGBD** Système de Gestion des Bases de Données. 1, 2, 14, 15, 17, 18, 21, 23, 24, 30, 59–62
- TLD** Top Level Domain, domaine de premier niveau. 35
- UAC** User Access Control, le système qui fait que Windows vous demande de confirmer certaines actions. 23
- WAMP** Windows Apache MySQL PHP. 10, 13
- WAN** Wide Area Network. 44, 48
- x64** 64 bits. 18–21, 53, 54, 60
- x86** 32 bits. 18–21, 53, 54, 60
- XHTML** eXtensible HyperText Markup Language. 7
- XML** eXtensible Markup Language. 7