

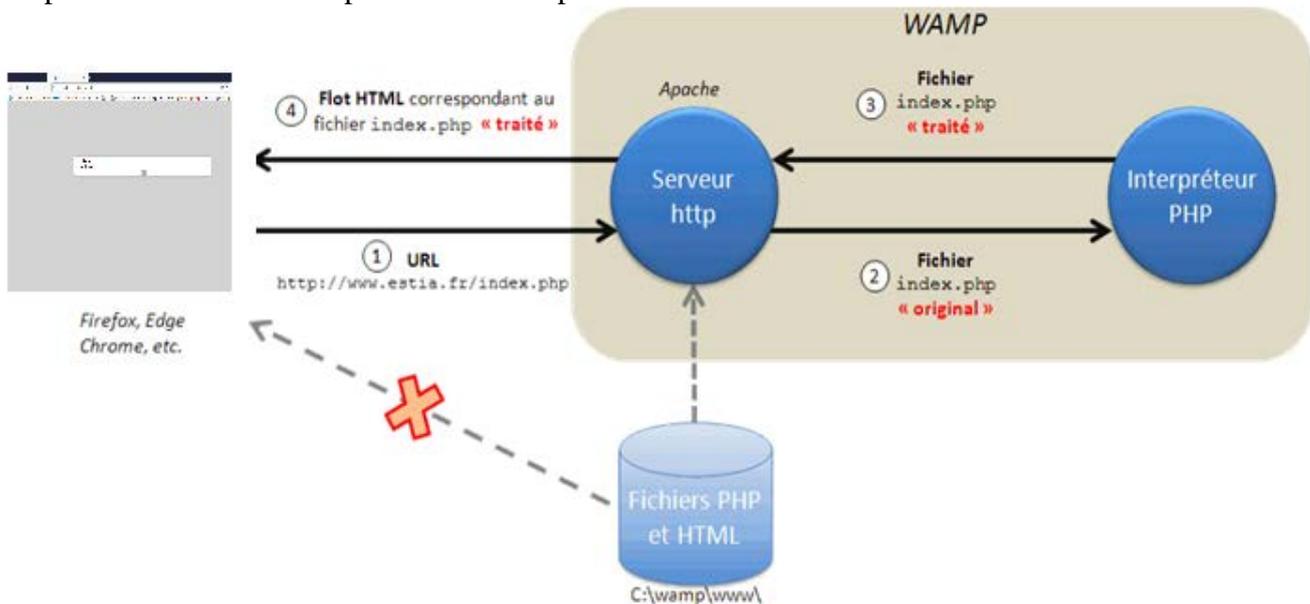
Objectifs : Ce premier TP propose de traiter les informations d'un formulaire d'authentification, les données utilisateurs étant contenues dans un fichier texte.

Source : <https://www.guillaumeriviere.name/estia/si/>

I. Première page PHP

I.1. Démarrer le serveur web :

Pour générer des pages web avec des scripts PHP, nous avons besoin : d'un serveur HTTP, d'un interpréteur PHP et d'un client HTTP. Ces trois éléments sont généralement installés sur des machines différentes. Dans le cadre de ce TP, nous utiliserons un paquetage (**WAMP** = **W**indows · **A**pache · **M**ySQL · **P**HP) qui comprend le serveur web Apache et un interpréteur PHP.

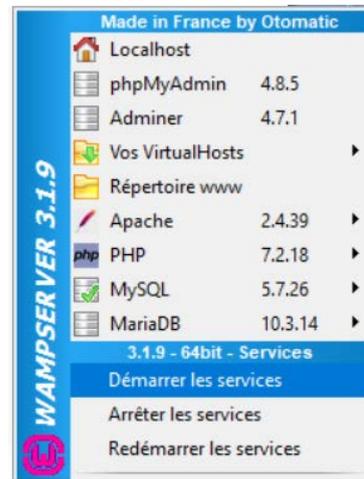


Lancer Wamp server :



Cliquer sur l'icône dans la barre des tâches, et faire démarrer les services

Quand le sigle "W" est vert, cela signifie que les serveurs sont en route.



- Créer un sous dossier TP_NSI dans le répertoire racine du serveur http

C : \wamp64 \www

- Créer un fichier nommé **test.html** et enregistrez le
- Éditez-le (avec SublimText par exemple) et recopiez le code suivant :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>page de test</title>
  </head>
  <body>
    <h1>test du serveur apache</h1>
  </body>
</html>
```

- Ouvrez-le dans un navigateur web en saisissant l'URL suivante :
- <http://127.0.0.1/TP-NSI/test.html>

1.2. [Fichier index](#)

2. Créez maintenant le fichier **index.php**, puis :

- Ajoutez un bloc de code PHP qui déclarera les trois variables ci-dessous et affichera le contenu de la troisième :

```
<?php
$date = date("d-m-Y");
$heure = date("H:i");
Print("Nous sommes le $date et il est $heure");
?>
```

- Ouvrez-le dans un navigateur web en saisissant l'URL suivante :

<http://127.0.0.1/TP-NSI/index.php>

- Faites un clic droit sur la page et dans le menu choisissez, selon le navigateur utilisé, "Afficher la source", "Code source de la page" ou encore "Code source". Que constatez-vous ?

II. [Accès sécurisé par formulaire](#)

Nous souhaitons restreindre l'accès de certaines pages à certains utilisateurs, authentifiés par mot de passe.

Pour ce premier TP, la liste des utilisateurs autorisés est stockée dans un fichier.

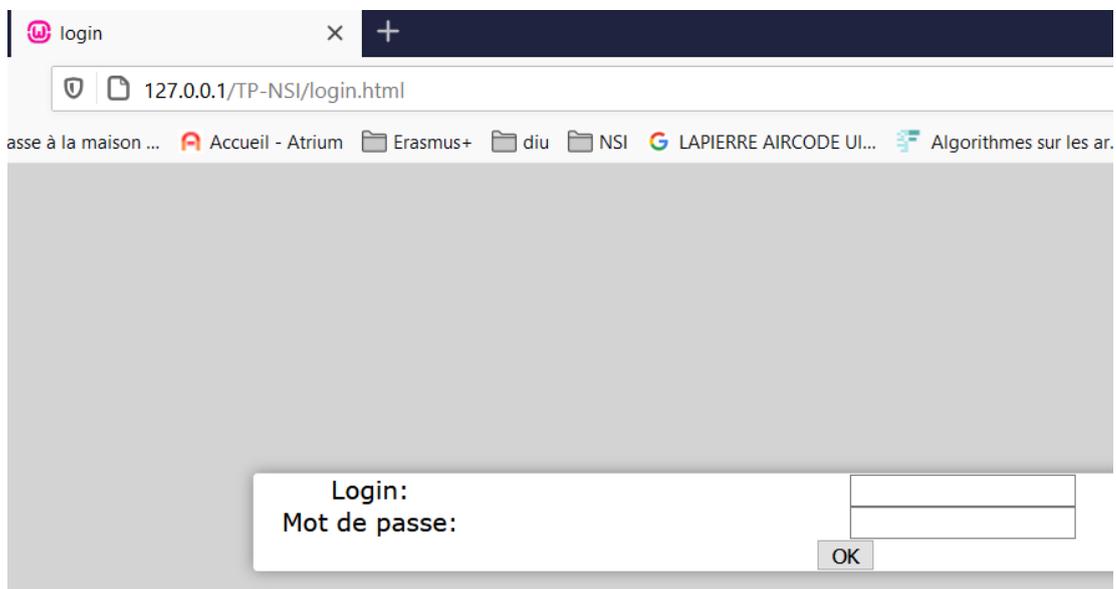
Toutes les connexions, et tentatives de connexion, sont enregistrées dans un *fichier de log*

II.1. [Saisie :](#)

1. Créez une page HTML nommée **login.html** comportant un formulaire.

Celui-ci comprend un champ "text" permettant à un utilisateur de saisir son "login", un deuxième champ "text" permettant à un utilisateur de saisir son "mot de passe", et un bouton de type "submit" permettant de valider les informations.

Ce formulaire doit appeler la page PHP nommée **valider.php**, que nous construirons plus tard. <http://127.0.0.1/TP-NSI/login.html>



II.2. Traitement

2. Créez un fichier nommé **valider.php**. Celui-ci représente une page Web qui affiche le message "Bienvenue XXXX", où XXXX est la valeur du login saisie par un utilisateur dans le formulaire de la page **login.html**.
3. Testez l'enchaînement entre la page **login.html** et le script **valider.php**.
1. Modifiez également le script **valider.php** de façon à ce que la page affiche un message d'erreur, et un lien de retour vers **login.html**, si le mot de passe saisi est différent de **331**, et continue à afficher "Valider ..." sinon.

II.3. Récupération d'informations par lecture d'un fichier

Nous souhaitons désormais stocker des mots de passe différents pour plusieurs utilisateurs. Pour cela, nous allons utiliser un fichier **mdp.txt** contenant les logins et mots de passe de tous les utilisateurs : chaque ligne contient un login et un mot de passe séparés par deux points ":". Chaque fin de ligne contient le caractère ":".

Par exemple :

mdp.txt
thomas:JimHend:
maud:HermGran:
arnaud:jarBin:
nathan:Neo:

1. Créez le fichier **mdp.txt**
2. Modifiez le script du fichier **valider.php** afin qu'il réalise l'authentification d'un utilisateur grâce à ce fichier.
 - o Le but est donc de regarder si le login de l'utilisateur est présent dans le fichier **mdp.txt**
 - o Vous aurez besoin de lire le fichier ligne par ligne, et d'analyser chaque ligne avec la fonction **explode()** pour vérifier si le login est celui recherché.
 - o Remarque : il se peut que le login demandé n'existe pas dans le fichier ! Il faudra donc gérer le cas où la fin de fichier est atteinte sans avoir trouvé le login dans le fichier.
3. Pour faire les choses proprement, modifiez votre code de manière à ce que la recherche du login dans le fichier soit effectuée par la fonction suivante (que vous allez créer) :

```
function getMdp ($login) {  
    ...  
}
```

Cette fonction **recevra en paramètre** une chaîne de caractères représentant le login de l'utilisateur recherché, **fera** la recherche du login dans le fichier **mdp.txt**, et **retournera** comme résultat le mot de passe associé à \$login ou bien une chaîne vide dans le cas où le login de l'utilisateur recherché n'est pas présent dans le fichier.

II.4. Stockage d'information par écriture d'un fichier

À chaque connexion ou tentative ratée de connexion, nous souhaitons enregistrer : le login, la date et l'heure, ainsi que le statut (réussie ou échouée) dans un fichier journal (aussi appelé fichier de « log »).

1. Modifiez le script du fichier **valider.php** afin qu'à chaque appel, il ajoute à un fichier **log.txt** les informations relatives à la tentative de connexion. Remarques :
 - Le mode d'ouverture "a+" de la fonction **fopen()** permet d'ouvrir un fichier en mode écriture tout en positionnant le curseur à la fin du fichier.
 - La fonction **fputs()** permet d'écrire dans un fichier.
 - La fonction **date()** permet de récupérer la date et l'heure.
2. Pour faire les choses proprement, modifiez votre code de manière à ce que l'écriture dans le fichier journal soit effectuée par la fonction suivante (que vous devez créer) :

```
function addLog ($login, $reussie) {  
    ...  
}
```

Cette fonction **recevra en paramètre** une chaîne de caractères \$login, représentant le login de l'utilisateur qui tente de se connecter, et un booléen \$reussie, qui vaudra **TRUE** pour signaler la réussite de connexion et **FALSE** pour signaler un échec, et **fera** l'ouverture et les écritures nécessaires dans le fichier **log.txt**.