

# Projet PHP

## TIDAL

Bruno Mascret

Antoine Gréa

Alain Becker

## ACRONYMS

## SUJET DU PROJET

Le projet que nous vous proposons porte sur la création d'un petit site d'e-commerce. Le but est d'afficher une liste de produits et de gérer un panier.

### Sujet de base

Votre site doit répondre en moins de une seconde, être accessible avec Chrome ou Firefox dans leur dernière version stable. Tout les menus doivent pouvoir être accessible en moins de 3 clicks.

Votre site doit également :

- Proposer une vitrine de produit sous forme de liste claire
- Permettre d'ajouter un produit à un panier
- Permettre la création d'un compte
- La consultation du panier
- La fusion de panier lorsque l'on se connecte ou l'on crée un compte.

### Demandes supplémentaires

Pour aller plus loin vous pouvez également :

- Améliorer la navigation avec des filtres, recherche et catégories de produits
- Identifier le pays et la langue du client automatiquement avec [GeoIP](#)
- Ajouter un captcha sur la page de création de compte

## Contraintes techniques

### Accessibilité et versions HTML

L'ensemble du site respectera les normes d'accessibilité de niveau AAA. Il sera écrit en HTML 5.

De plus, toute page du site devra être accessible de la page d'accueil en 3 interactions au maximum (clic, raccourci, etc.).

La navigation devra être possible sans souris ou sans clavier.

### Environnement d'exécution

Le site devra être affichable et utilisable à partir d'un navigateur Chrome à jour.

La partie dynamique (serveur) devra fonctionner sur une architecture PHP/MySQL orienté REST.

Toutefois, vous restez libres des outils et environnements de développement que vous souhaitez utiliser pour arriver à ce résultat.

### Base de donnée

La base de donnée vous est fournie sous la forme d'un script SQL de génération de base de données.

Vous êtes libre de modifier la base dans la limite de conserver les données existantes.

### Graphisme

La partie graphique doit avant tout éviter les erreurs d'accessibilité et d'ergonomie.

Il ne vous est pas demandé de faire un site esthétiquement joli, mais simple et fonctionnel.

S'il vous reste du temps en fin de module, vous pourrez améliorer cet aspect à ce moment-là.

## Contraintes de réalisation

Toutes les pages doivent être consultables en basse résolution et disposer d'une mise en page spéciale pour l'impression.

## Page d'accueil

La page d'accueil doit disposer d'un menu et d'un formulaire d'identification pour les utilisateurs. On y retrouve aussi la liste des produits les plus consultés (ajouté au panier).

## Autres pages

Vous êtes totalement libre de l'organisation des autres pages (nombre, architecture, etc...).

Vous réaliserez cependant une page d'information présentant les développements que vous avez réalisés, vos sources, les auteurs, ainsi que les ressources bibliographiques et la webographie que vous avez utilisé.

## Fonctionnalités à implémenter

### Liste des produits

Vous réaliserez une liste des produits qui soit simple à consulter. Chaque produit peut être ajouté au panier.

### Compte utilisateur

Vous proposerez un système de gestion des utilisateurs (inscription, login, session, etc.).

## Rendu

À la fin du projet, on doit obtenir :

- Un dépôt Github avec tout le code source du site web.
- Une documentation de votre code source en utilisant phpdocumentor.
- Un compte-rendu de votre travail, de vos sources et de vos difficultés.
- Une fiche d'auto-évaluation disponible sur l'eCampus.

### Note

Ne pas oublier :

- de citer vos sources (vraie bibliographie)
- de vérifier que vous avez le droit d'utiliser les ressources dans votre site.
- de maintenir votre webographie en fonction de vos utilisations de ressources.

# INSTALLATION D'UN SERVEUR WEB COMPLET

Dans cette section, nous détaillons le processus d'installation d'un serveur web. Pour le projet nous avons besoin d'un serveur Apache 2 avec un interpréteur PHP et une base de donnée MySQL. Pour simplifier l'accès à la base de donnée, le logiciel PHPMyAdmin est inclus.

## Installation simplifiée sur Windows

Sur Windows il est préférable d'installer WAMP. Vous pouvez retrouver le projet principal sur [wampserver.com](http://wampserver.com).

Il est également de nombreuses variantes, certaines portable (ne nécessitant pas d'installation) comme [XAMPP](#).

### Note

Étant donné la variété et complexité des installation de type WAMP, aucun support supplémentaire ne sera donné pour ce mode d'installation.

Pour ce projet le moyen préféré est l'utilisation d'une machine Ubuntu.

## Installation sous Ubuntu

Sous Ubuntu il faut utiliser le terminal et entrer les commandes suivantes :

```
sudo apt install apache2 php php-pear \  
libapache2-mod-php mysql-server \  
php-mysql phpmyadmin  
sudo apt install php-curl php-gd \  
php-intl php-json php-mbstring \  
php-xml php-zip
```

Ceci installe un serveur Apache 2, la base de donnée MySQL, un interpréteur PHP ainsi que PHPMyAdmin. La deuxième commande installe des modules classiques pour PHP.

Pour installer des modules supplémentaire dans PHP il faut soit utiliser `apt install` avec le paquet correspondant à la bibliothèque souhaitée soit utiliser `pear`.

Par exemple pour installer le générateur de documentation PHP il faut utiliser `sudo pear channel-discover pear.phpdoc.org && sudo pear install phpdoc/phpDocumentor`.

### Avertissement

Ces manipulations demandent des droits d'administrateur et ne doivent pas être faites sur les installations de l'école.

## Utilisation de la Machine Virtuelle

Pour simplifier ce processus, une machine virtuelle Ubuntu avec l'installation complète est fournie sur [cpexplorer](#).

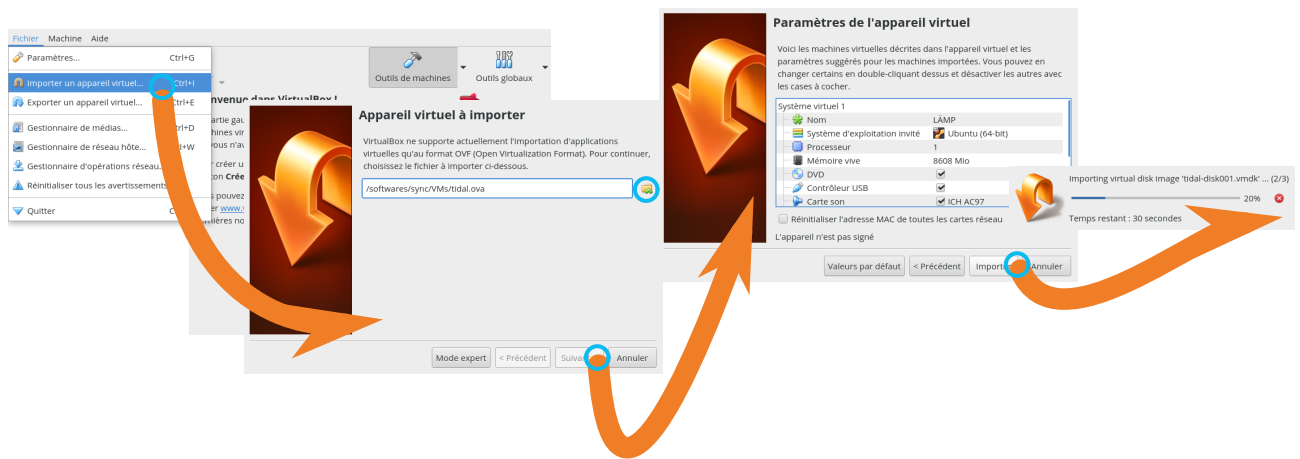
Ce fichier est normalement synchronisé depuis le montage réseau `/softwares/sync/VMs/tidal.ova` vers le disque dur de la machine dans `/sync/VMs/tidal.ova` mais la synchronisation peut être défectueuse. Dans ce cas vous pouvez soit importer depuis le montage réseau directement ou depuis le fichier téléchargé sur cpexplorer.

### Avertissement

Voici les problèmes possibles sur vos postes et leurs solutions :

- VirtualBox produit une erreur au lancement → supprimez le dossier de configuration `~/.config/VirtualBox`
- Le fichier `.ova` est indisponible dans `/sync/VMs/tidal.ova` → importez depuis `/softwares/sync/VMs/tidal.ova`.
- Le fichier `.ova` est indisponible dans `/softwares/sync/VMs/tidal.ova` → télécharger sur cpexplorer.
- Le dossier `/var/tmp` est inaccessible → raportez le problème au mail [support@cpe.fr](mailto:support@cpe.fr) et à votre encadrant en indiquant le nom du poste puis changez de poste.
- Une erreur survient à l'importation → supprimez le dossier `/var/tmp/VMs/LAMP` et recommencez.

Importez la machine dans VirtualBox en utilisant le menu **Fichier > Importer un appareil virtuel...**



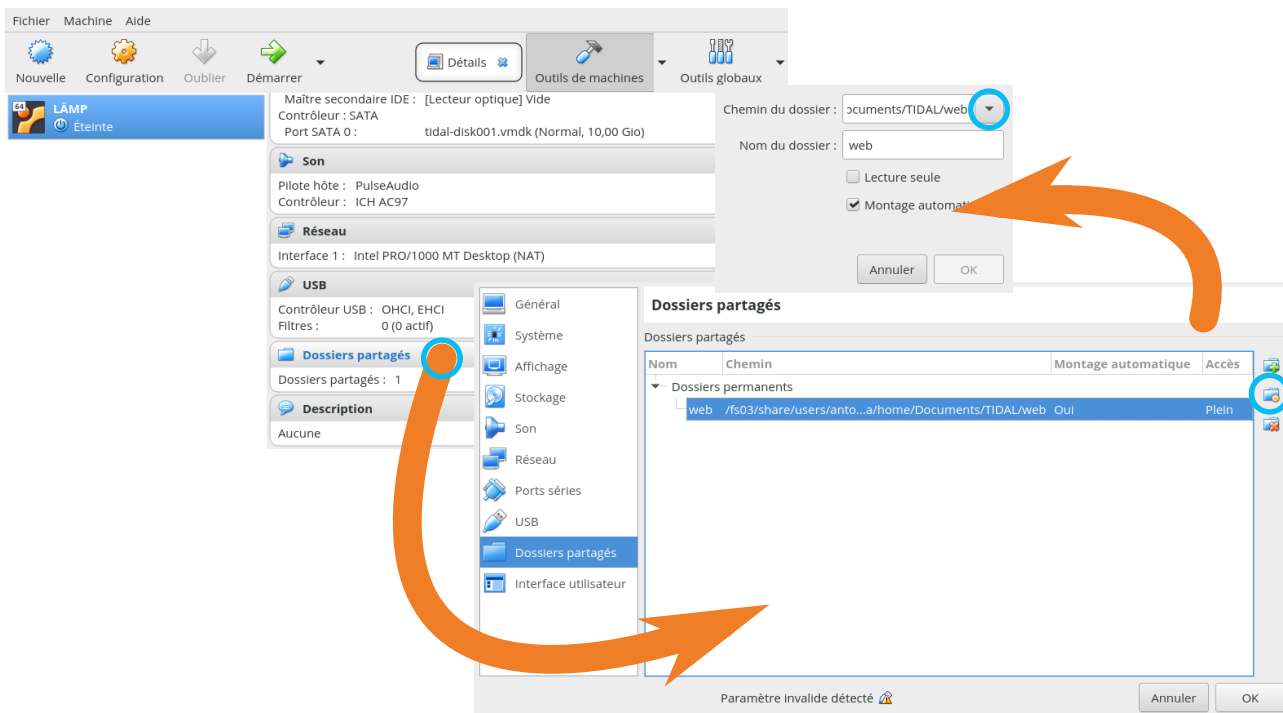
**Figure 1:** Procédure d'importation VirtualBox.

### Note

Cette machine est un Ubuntu Server avec les caractéristiques suivantes: (host=votre machine, guest=machine virtuelle)

- OpenSSH Serveur
  - SSH: 22(guest) → 2222(host)
  - Accès host: `ssh cpe@localhost -p 2222`
  - Utilisateur cpe
  - Mot de passe tp
- Apache 2
  - HTTP: 80(guest) → 8080(host)
  - Root Web : web(partage VirtualBox) → `/var/www`
  - Configuration du site : `config/default-site.conf`(partage Virtual-Box) → `/etc/apache2/sites-enabled/000-default.conf`
  - Accès host: <http://localhost:8080>
- PHP 7
  - Configuration : `config/php.ini`(partage VirtualBox) → `/etc/php/7.4/apache2/php.ini`
- MySQL server
  - Utilisateur debian-sys-maint
  - Mot de passe aR7RIRZbiUZw3dYk
- PHPMYAdmin
  - Accès host: <http://localhost:8080/phpmyadmin>

Créez un dossier partagé nommé web qui pointe sur votre code source. Celui-ci sera monté dans `/var/www`. Les fichiers dans ce répertoire auront les droits `rxw` pour tous afin d'être accessible au serveur web dans la machine virtuelle.



**Figure 2:** Changement du partage réseau.

### Avertissement

Ce genre de pratique avec les droits ne doit jamais être fait en **production**. Pensez à rendre votre environnement de production bien plus robuste que celui de développement. Aussi celui-ci doit être créé avec rigueur et expertise. Séparez les différents processus et services et surtout protégez vos réseau avec des **proxy et pare-feu matériels**.

Vous pouvez vous connecter à la machine virtuelle depuis votre poste en utilisant SSH et SFTP: `ssh cpe@localhost -p 2222` avec le mot de passe tp.

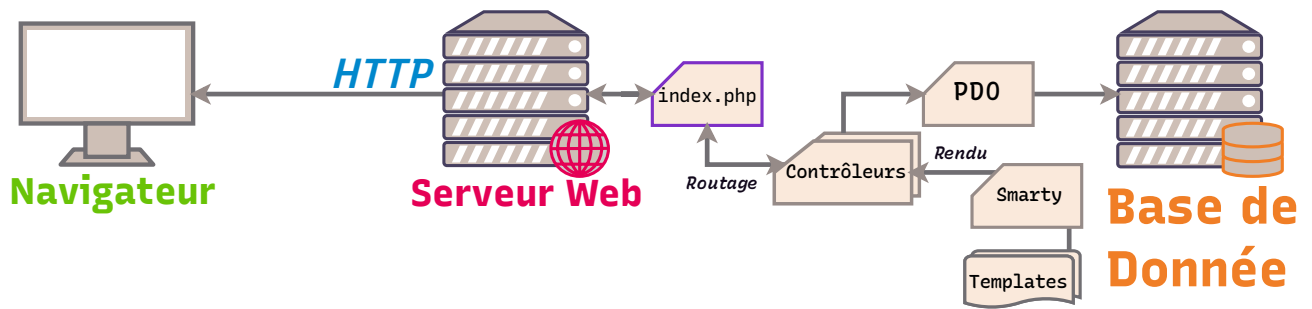
### Importation de la base de donnée

Dans l'e-Campus, vous trouverez une archive avec une architecture du projet. Dans le dossier web/bd vous aurez un fichier SQL qui sera à importer. Utilisez PH-PMYAdmin pour plus de simplicité. La base de donnée est accessible en utilisant le nom d'utilisateur debian-sys-maint et le mot de passe ar7RIRZbiUzW3dYk.

## ARCHITECTURE

Vous veillerez à anticiper l'organisation physique des scripts et des ressources sur votre serveur (arborescence, droits, etc.).

Le projet doit respecter l'architecture suivante au plus près.



**Figure 3:** Architecture du projet de site PHP.

On souhaite avoir une séparation MVC maximale donc privilégiez REST.

Voici un exemple de fichier `index.php` principal :

```

<?php
session_start();
require('./libs/smarty/Smarty.class.php');
require('./controllers/controller.php');
require('./models/model.php');

try {
    $bdd = new PDO("mysql:host=localhost;dbname=webapp",
        "debian-sys-maint", "aR7RIRZbiUZw3dYk");
    $bdd->query("SET NAMES UTF8");
} catch (Exception $e) {
    echo "Problème de connexion à la base de donnée !";
    die();
}

page = route($URL);
smarty->display('./views/home.tpl');
```

## Ébauche statique du site

Commencez par créer un site web statique avec des données figées. Vous pouvez vous arrêter à la liste des premières fonctionnalités *inclues*. Vous penserez notamment à prévoir les réactions aux événements, et prendrez soin de découvrir les fonctionnalités nécessaires mais non exprimées directement par le client.

Vous réaliserez notamment les formulaires, qui seront pour le moment passifs (il n'y aura pas d'action côté serveur lorsqu'ils seront validés).

Vous pouvez cependant déjà prévoir et réaliser des contrôles javascript.

**Attention aux pertes de temps !** Nous ne vous demandons pas de maîtriser en quelques heures HTML et CSS !

Le temps passe souvent vite quand on cherche à régler de petits détails. C'est formateur, mais il ne faut pas oublier que vous êtes en temps limité.



Profitez donc au maximum de la présence des enseignants qui pourront vous dépanner ! Cherchez d'abord une solution, et si vous ne trouvez rien au bout de 15 minutes, sollicitez-les !

## Formulaires Dynamique

L'objectif est d'utiliser des formulaires pour :

- Envoyer des paramètres pour construire des pages dynamiques sur les produits (filtrage, recherche...)
- Gérer la connexion de l'utilisateur.

Ces formulaires seront accessibles (HTML, CSS et ARIA).

Ils pourront utiliser AJAX pour rendre l'utilisation du site plus conviviale et ergonomique.

## Modèles

L'objectif de ce TP est de rendre votre site dynamique en utilisant PHP.

Vous utiliserez un système de templating pour votre site. On vous recommande [Smarty](#) avec ce tutoriel disponible à l'adresse [fredods.com/creer-votre-site-avec-smarty-en-mvc-partie-1](https://fredods.com/creer-votre-site-avec-smarty-en-mvc-partie-1).

## Routage

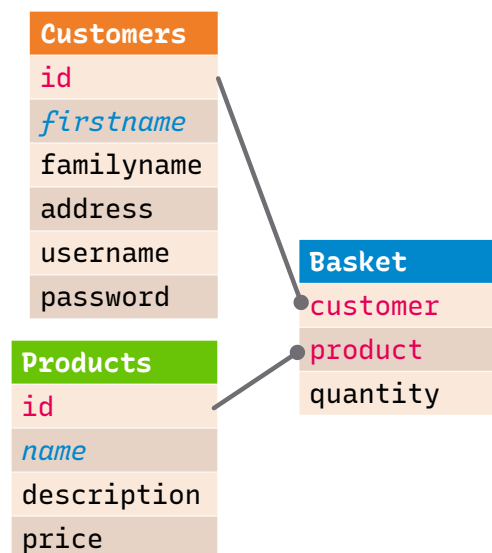
Pour faire une architecture MVC, il faut adopter le routage dans votre projet. Vous pouvez consulter le site à l'adresse [openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4682351-creer-un-routeur](https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4682351-creer-un-routeur) pour plus de détails.

## Base de donnée

Pour accéder à la base de donnée depuis PHP il faut utiliser la bibliothèque PDO. Vous trouverez des ressources utiles sur le sujet sur [ce site](#).

Pensez à réaliser :

- Une classe par entité de base de donnée
- Des requêtes préparées
- Des connexions sécurisées
- Vous protéger contre les [injections SQL](#)



**Figure 4:** Schéma de la base de donnée.