

# Projet : LeCoinSport

Dossier technique

NDZOUBA Sylvestre – BRISSON Dorian – MBOMBO  
MULUNDA Hence  
13/11/2022

## Sommaire

<b>Projet.....</b>	<b>2</b>
Contexte .....	2
Clients cibles.....	2
Cahier des charges.....	2
<b>Environnements de développements .....</b>	<b>3</b>
Présentation du Framework.....	3
Description des différentes technologies utilisées .....	3
Procédure d'installation .....	4
<b>Architecture et conception .....</b>	<b>6</b>
Modèle de données.....	6
Découpage des controllers.....	6
Présentation du templating .....	7
Gestion de la sécurité et des rôles utilisateurs .....	8
Solution d'upload des fichiers .....	8
Structure des fichiers de style .....	8
Structure des fichiers JS.....	8

# Projet

## Contexte

LeCoinSport est un site de petites annonces entre particuliers, directement inspiré du site internet LeBonCoin. Le principe peut se comprendre en une sorte de “vide grenier” en ligne où les utilisateurs peuvent vendre ou acheter un produit en lien avec le sport. L’idée principale du site réside dans la proximité des échanges entre utilisateurs. On y trouve une diversité d’annonces qui va des maillots de basket aux chaussures de course tout en passant par les lunettes de piscine, à titre d’exemple.

## Clients cibles

Ce site de vente en ligne est accessible à tout le monde, c’est-à-dire aussi bien aux particuliers qu’aux professionnels. Le site met en commun les annonces classées par catégories des produits et par type.

## Cahier des charges

- Inscription / Connexion / Déconnexion
- Ajouter produit
- Réserver un produit
- Acheter un produit
- Consulter les produits
- Consulter infos personnelles
- Modifier infos personnelles
- Contact

Espace administrateur (administrateur) :

- Ajouter un compte
- Supprimer un compte
- Bloquer un compte
- Changer le rôle d’un utilisateur
- Supprimer produit
- Ajouter produit
- Ajouter un acheteur
- Supprimer un acheteur
- Ajouter une catégorie
- Supprimer une catégorie
- Ajouter une commande

- Supprimer une commande
- Ajouter une ville
- Supprimer une ville
- Ajouter contact
- Supprimer contact
- Ajouter confirmation livraison
- Supprimer confirmation livraison
- Ajouter moyen de paiement
- Supprimer moyen de paiement

Espace administrateur (intermédiaire) :

- Ajouter un compte
- Supprimer un compte
- Bloquer un compte
- Changer le rôle d'un utilisateur
- Supprimer produit
- Ajouter produit
- Ajouter un acheteur
- Supprimer un acheteur

## Environnements de développements

### Présentation du Framework

Pour ce projet nous avons utilisé Symfony, qui est un framework qui représente un ensemble de composants PHP qui peuvent être utilisés dans des projets web privés ou open source. Également, c'est un **puissant Framework PHP** développé par une société française : **SensioLabs**. Il permet de réaliser des sites internet dynamiques de manière rapide, structurée, et avec un développement clair.

### Description des différentes technologies utilisées

Pour le développement de ce projet, nous avons dû utiliser différentes technologies :

- **HTML** : Le langage HTML est un **langage de description** sujet à interprétation. Le principe du langage est de décrire **la structure et le contenu** d'une page. Il va permettre de définir les différentes zones d'affichages (en-tête, corps de la page, pied de page, menu, etc) et d'intégrer son contenu (texte, image, tableau, vidéo, etc).

- CSS : Le langage CSS est destiné à gérer le contenu et la structure d'une page web. Il est techniquement possible de faire de la mise en forme en HTML mais les possibilités restent très limitées. Le langage **CSS** est donc là pour étendre considérablement les possibilités graphiques. L'acronyme CSS signifie **Cascading Style Sheet** autrement dit en français, **feuille de styles en cascade**, cela indique bien que les CSS permettent de styler, de mettre en forme une page web.
- Bootstrap : Bootstrap est un framework CSS. C'est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble et qu'on va pouvoir utiliser pour créer des design complexes de manière relativement simple.
- PHP : PHP (PHP : Hypertext Preprocessor) est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP est donc un langage que l'on va principalement utiliser pour générer des pages Web et créer des sites Web dynamiques.
- Twig : Twig est un moteur de templates développé par Sensiolabs (qui a aussi développé le Framework PHP Symfony). Le rôle d'un moteur de templates est principalement d'aider dans la lisibilité et la logique du projet en général et du code en particulier.

## Procédure d'installation

Pour faire fonctionner Symfony sur son ordinateur, il y a quelques prérequis. Il faut dans un premier temps installer un serveur local (Wamp / Xampp / Lamp / Mamp ) et installer également Git (<https://git-scm.com/>), NodeJS (<https://nodejs.org/en/>) et le gestionnaire de dépendances Composer (<https://getcomposer.org/download/>). Enfin, il faut évidemment installer Symfony (<https://symfony.com/download>). Une fois Symfony installé, il faudra extraire le fichier .zip dans un dossier système (ex : C:\Program Files\Symfony). Depuis le menu Démarrer, ajouter aux variables d'environnement le dossier dans lequel se trouve symfony.exe. Il faut maintenant ouvrir un invite de commande dans le dossier du serveur local (Pour Wamp: C:\wamp\www, pour Xampp : C:\xampp\htdocs). Ensuite, dans le dossier du projet sur Github, vous trouverez le fichier **lecoinsport.sql**. Il faut alors créer une base de données qu'il faut nommer lecoinsport. Il ne vous reste plus qu'à importer le fichier SQL pour que la base de données soit complète. Par la suite, créer un projet vide avec la commande `symfony new mon_projet--version="6.1.*" -webapp`

Modifier le fichier .env : Si vous utilisez MariaDb, modifier le DATABASE\_URL comme ceci :

```
DATABASE_URL="mysql://username:password@127.0.0.1:3306/lecoinsport?serverVersion=mariadb-X.X.X&charset=utf8mb4"
```

X.X.X doit être remplacé par la version de mariaDB sur votre ordinateur (ouvrir PhpMyAdmin pour connaître la version).

username:password doit être remplacé par vos logins de connexion à la base de données (en général il suffit de mettre root: ).

Si vous n'avez pas réussi à installer la commande symfony, et donc à créer un projet vide comme ci-dessus, utilisez composer :

- `composer create-project symfony/skeleton:"6.1.*"`  
    `my_project_directory`
- `cd my_project_directory`

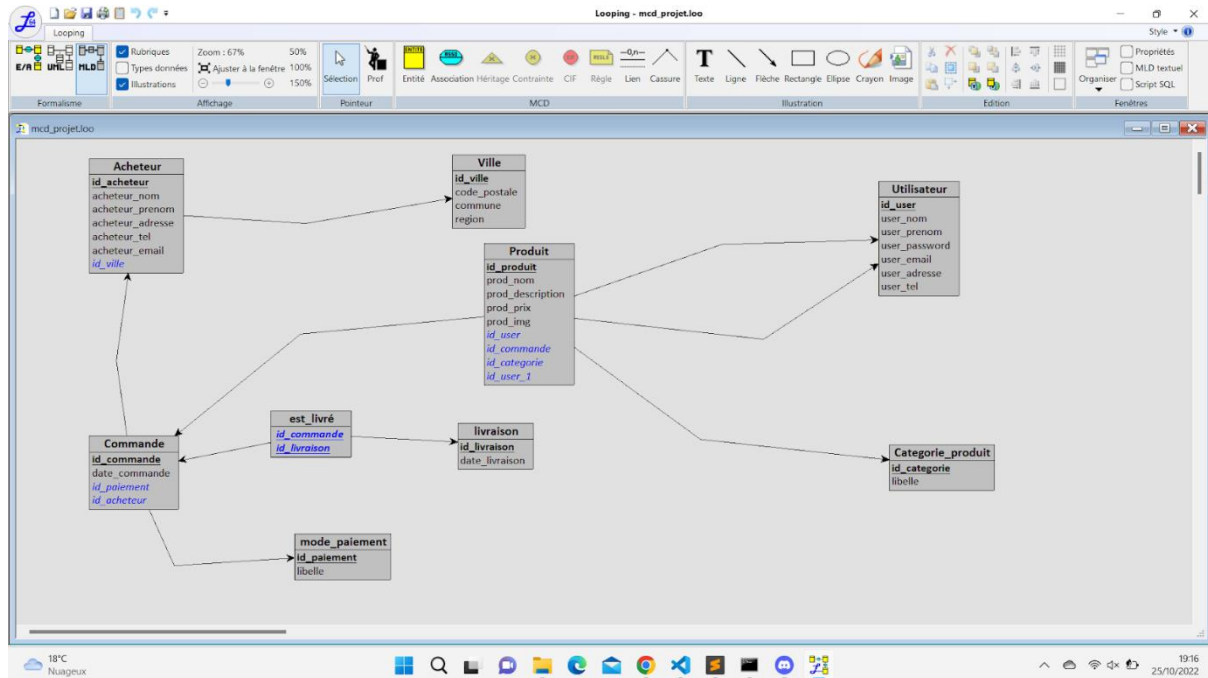
Pour importer le fichier sur votre ordinateur, il suffit de vous placer dans le dossier à l'aide de la commande `cd my_project_directory` et d'effectuer la commande `git clone https://github.com/SylvestreNdzouba/LeCoinSport/tree/master` .

Si la méthode précédente ne fonctionne pas, copiez tous les fichiers présents dans le dossier du projet sur Github et collez-les dans votre projet vide (hormis le fichier .env que vous avez configuré au préalable)

Il ne vous reste plus qu'à lancer le serveur à l'aide de la commande `symfony serve` et le tour est joué !

# Architecture et conception

## Modèle de données

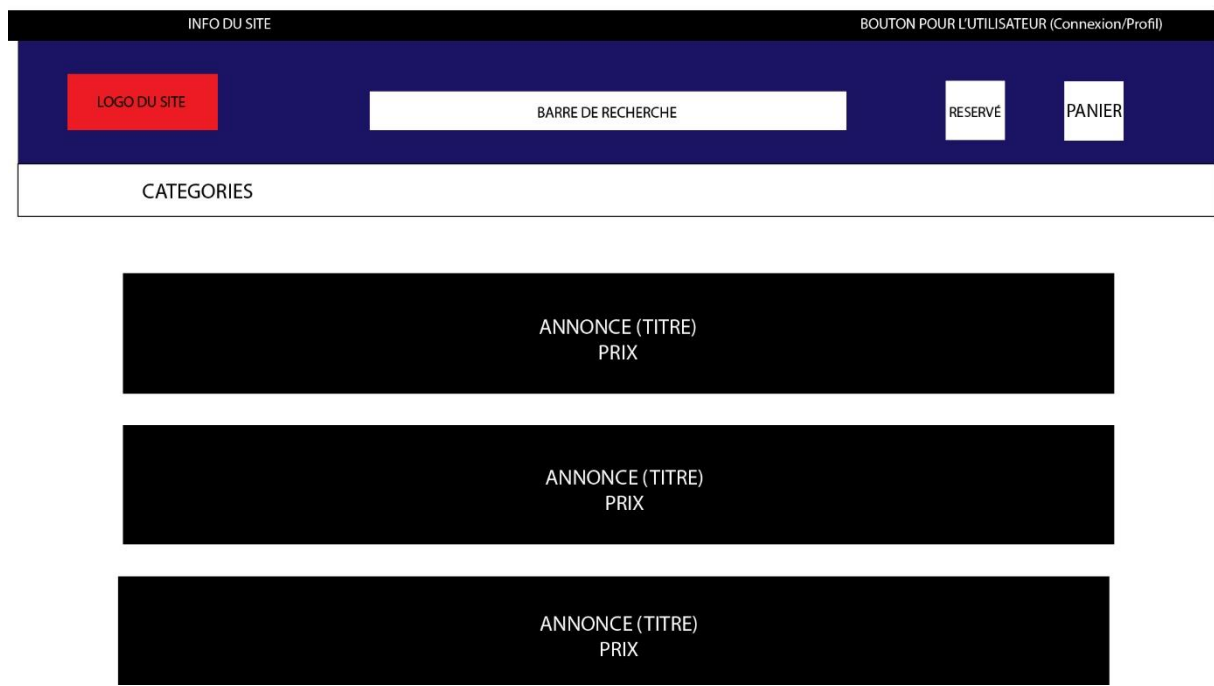


## Découpage des controllers

- AcheteurController : Ce controller permet d'ajouter un acheteur à la base de données, on peut également accéder à ses informations.
- AnnonceController : Ce controller permet de voir toutes les annonces présentes sur le site LeCoinSport.
- CategorieProduitController : Ce controller permet d'ajouter/modifier des catégories, il est réservé aux administrateurs du site.
- CommandeController : Ce controller permet de créer une commande lorsqu'un utilisateur veut acheter un produit.
- ContactController : Ce controller permet d'envoyer un message aux administrateurs du site lorsqu'un utilisateur a par exemple une question relative au site, ou lors d'un litige d'une commande.
- EstLivreController : Ce controller indique si lors d'une vente le produit à bien été livré à l'utilisateur acheteur.
- LivraisonController : Ce controller permet de suivre l'avancement d'une livraison lors d'un achat effectué par un utilisateur.
- ModePaiementController : Ce controller permet de sélectionner un mode de paiement lors d'un achat.

- **ProduitController** : Ce controller référence tous les produits vendus sur le site et permet de créer un produit lorsqu'un utilisateur veut vendre un bien.
- **RegistrationController** : Ce controller permet aux utilisateurs de s'inscrire sur le site pour pouvoir accéder à certaines fonctionnalités supplémentaires lors d'un achat. Il est également obligatoire de s'inscrire si l'on veut vendre un produit.
- **SecurityController** : Ce controller gère la connexion d'un utilisateur au site, il faut qu'il soit au préalable inscrit.
- **UsersController** : Ce controller permet aux utilisateurs de voir leurs propres informations. Ils peuvent également modifier leurs informations personnelles.
- **HomeController** : Ce controller gère la page d'accueil.

## Présentation du templating





## Gestion de la sécurité et des rôles utilisateurs

Sur le site LeCoinSport, les utilisateurs anonymes sont considérés comme non identifiés, ils ont alors droit à moins de fonctionnalités sur le site. Par exemple, ils ne peuvent pas réserver un produit (cette fonctionnalité permet d'être prioritaire sur l'achat). En ce qui concerne les utilisateurs connectés au site, puisqu'ils sont considérés comme membres, ils ont droit à ces mêmes fonctionnalités. Sous Symfony, c'est le firewall qui prend en charge l'authentification, il permet une connexion sécurisée au site. De plus, grâce à celui-ci, certaines pages du site peuvent être réservées (comme sa propre page de profil) sans que d'autres utilisateurs puissent y accéder. Autrement dit, il faudra que le visiteur soit authentifié pour que le firewall l'autorise à passer. L'autorisation intervient après l'authentification. Comme son nom l'indique, c'est la procédure qui va accorder les droits d'accès à un contenu. Sous Symfony, c'est l'access control qui prend en charge l'autorisation.

## Solution d'upload des fichiers

Concernant l'upload de fichier, les utilisateurs qui souhaitent vendre un produit ont la possibilité d'ajouter une photo à leur annonce. Pour se faire, nous avons au préalable enregistré dans la base le champ image au type blob, qui est un type d'enregistrement d'image. Ensuite, il nous a suffi d'utiliser FileField dans le controller de produit qui permet d'aller chercher des fichiers enregistrés dans l'ordinateur de l'utilisateur. Il n'aura plus qu'à sélectionner l'image qui lui convient.

## Structure des fichiers de style

Pour les fichiers de style, les fichiers CSS sont rangés dans public/assets/css, où on y retrouve un thème bootstrap et des styles pour des pages bien précises. Ces fichiers sont ensuite appelés dans base.html.twig qui est situé dans le dossier template. Tous les fichiers twig héritent de base.html.twig et donc tous les styles et le thème bootstrap sont eux aussi appelés.

## Structure des fichiers JS

Pour les fichiers JS, ces derniers sont rangés dans public/assets/js, on y retrouve la suite du thème bootstrap mais pour les animations JS. Ces fichiers sont ensuite appelés dans le même principe que pour les fichiers de style.