Rapport de Projet

Membres du projet :

- Baptiste Saint-Léger
- Valdrin Salihi

Année : 2021-2022

TABLE DES MATIÈRES

Localisation du site & SEA

Localisation du site :

L’adresse du site web n’est ni installée sur tpweb ni sur une adresse publique

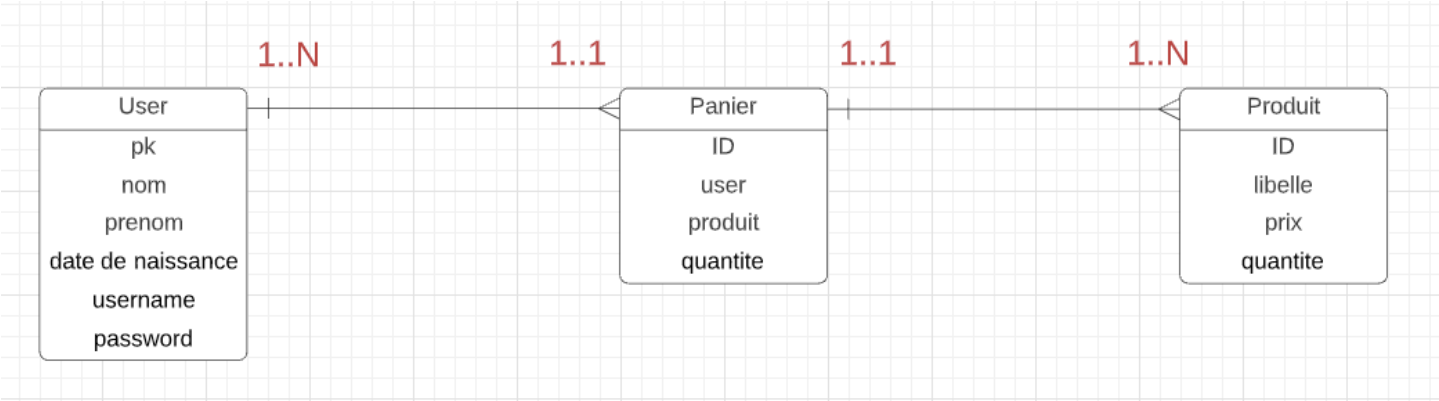


Schéma entité association (SEA) de notre base de données

Organisation du code

Hiérarchie des classes dans les répertoires :

Nous avons pris le réflexe au sein de chaque répertoire du projet de nommer les fichiers de la manière la plus expressive possible afin que lorsque l'on regarde d'un point de vue extérieur l'on puisse s'y retrouver rapidement. Ici, nous allons voir une liste des répertoires et classes que nous avons estimée intéressant à montrer : (*ps : les répertoires sont soulignés afin de les dissocier des fichiers*)

projet :

- config :
- database :
- public :
- src :
- templates :

- Au sein de database nous avons un unique fichier du nom de mybase.db qui est la base de données de notre site et qui va nous permettre de stocker toutes les tables de chacune de nos entités.

- Dans public nous avons un autre répertoire nommé CSS qui contiendra deux fichiers .css, un pour le style graphique de toutes les pages de notre site et un autre fichier qui s'occupera du style de la police du site en entier.

- Pour le répertoire source (src) nous allons encore le décomposer, car il contient plusieurs sous-répertoires :

src :

- Controller : AdminController.php, ProduitController.php, SecurityController.php,

  SiteController.php, SuperAdminController.php, UserController.php.

- Entity : Panier.php, Produit.php, User.php .
- Form : EditUserType.php, ProduitType.php, RegistrationType.php .
- DataFixtures : AppFixtures.php
- Repository : PanierRepository.php, ProduitRepository.php, UserRepository.php
- Security : MyAuthenticator.php
- Service : total.php

Organisation du code (bis)

Et en dernier répertoire, mais pas des moindres, nous avons le templates qui va s'occuper de contenir toutes les vues de notre site et il se compose ainsi :

templates :

- admin : ajouterProduit.html.twig, users.html.twig .
- base : base.html.twig, base-vide.html.twig .
- menu : admin.html.twig, anonymous.html.twig, superadmin.html.twig, users.html.twig .
- produit : affichePanier.html.twig, productList.html.twig .
- security : login.html.twig, registration.html.twig, registrationAdmin.htm.twig.
- superadmin : admins.html.twig, editSuperAdmin.html.twig
- user : edituser.html.twig, panier.html.twig .

Quelques précisions sur le code :

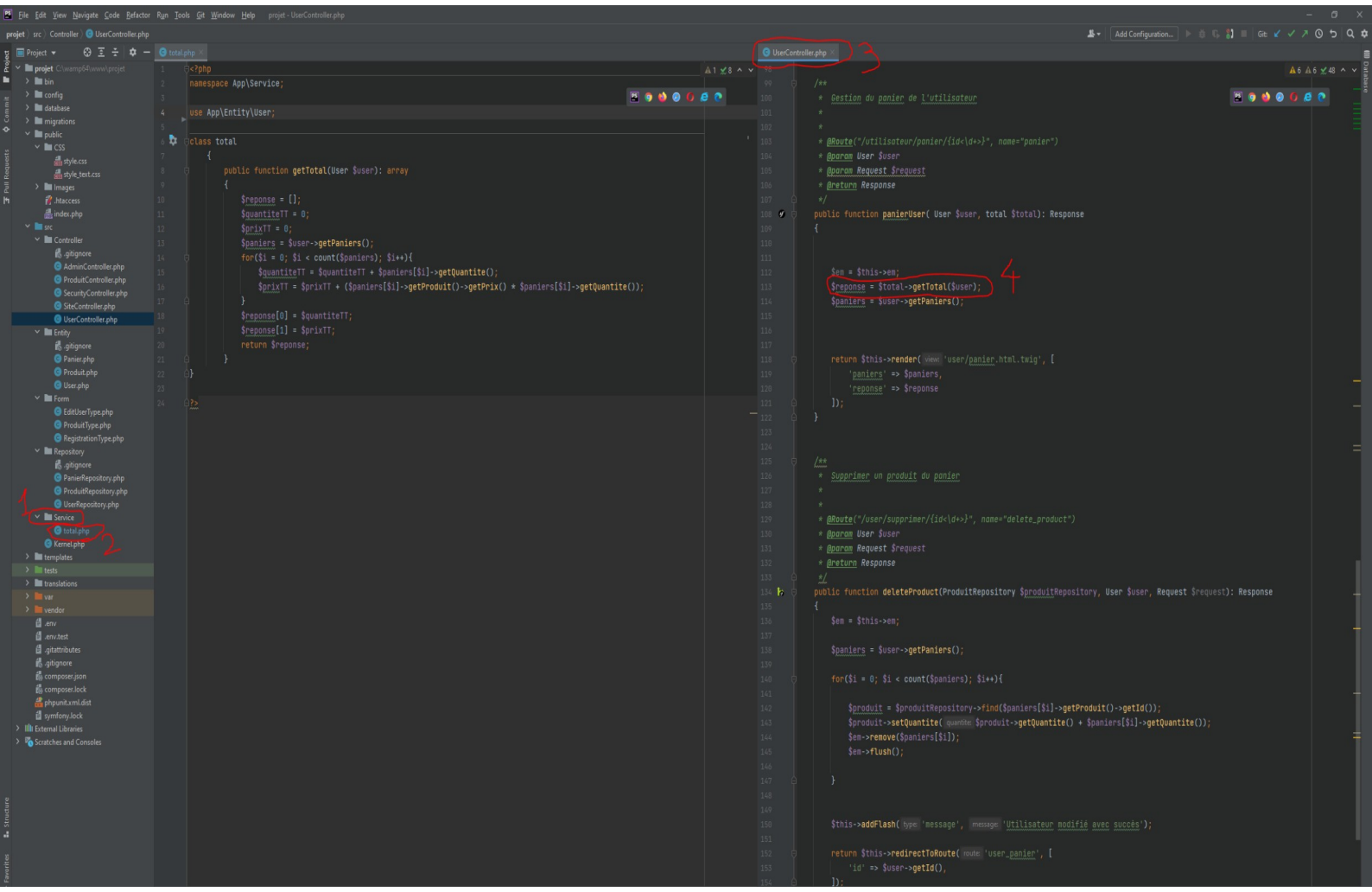
Que ce soit les controllers, les entités ou bien les formulaires, ils ont tous étaient crées à partir de symfony. Dans le Controller, nous avons bien dissocié les actions pour la partie administrateur, la partie client, celles des produits, de la sécurité et du site dans son ensemble.

Au sujet des templates nous sommes partie du principe de faire un squelette général pour l'ensemble des vues du nom de « base.html.twig » afin que la quasi-totalité de nos vues parte de ce point-là et rajoute les éléments précis dont on a besoin au cas par cas.

Les controllers vont permettre de créer les actions de chaque composant essentiel au site (différents types d'utilisateurs, aspect sécurité, produits, etc..). Les entités quant à elles régiront les attributs et méthodes de chacunes de nos tables (produit, panier, user). Les templates seront le pont visuel entre les actions de nos controller et l'interface de l'utilisateur.

Création d’un service

Tutoriel pour créer un service :



1. Créer un sous-répertoire du nom de Service dans le répertoire src .
2. Ensuite créer un fichier du nom du service que vous désirez avec une extension .php contenant les actions voulu pour ce service.
3. Une fois le service crée, il faut le brancher au controller dont il sera utile sans oublier de bien injecter la dépendance du service dans le namespace du controller (../Service/service.php) .
4. Enfin faire appel à la fonction du service dans l’action lié dans le controller .

Points Particuliers

À l’issue de ce projet, nous avons constaté principalement une erreur de fonctionnement de l’action de l’ajout de produit qui crée des doublons dans le panier utilisateur. Le problème consiste au fait que l’on injecte de nouveaux « paniers » à chaque fois dans le panier principal au lieu d’injecter une liste de produits permettant ainsi d’éviter le dédoublement de ligne et tout additionné dans une seule même ligne. Et le dernier aspect différent du fonctionnement attendu pour la partie administrateur est qu’il ne peut voir les super administrateur car l’on pensait qu’il n’avait pas de raison de le voir étant donné le fait qu’il est un cran en dessous en terme de droit. En revanche le super administrateur lui peut voir les administateurs, en créée de nouveaux et en supprimer. C’est donc un principe de chaîne que l’on a voulu instaurer de Client → Administrateur → Super Administrateur.

REMERCIEMENT