

SYMFONY

L3 - Informatique / Université de Poitiers



KORTSMIT Simon - PORRO Guillaume

29/04/22

Technologies du Web 2

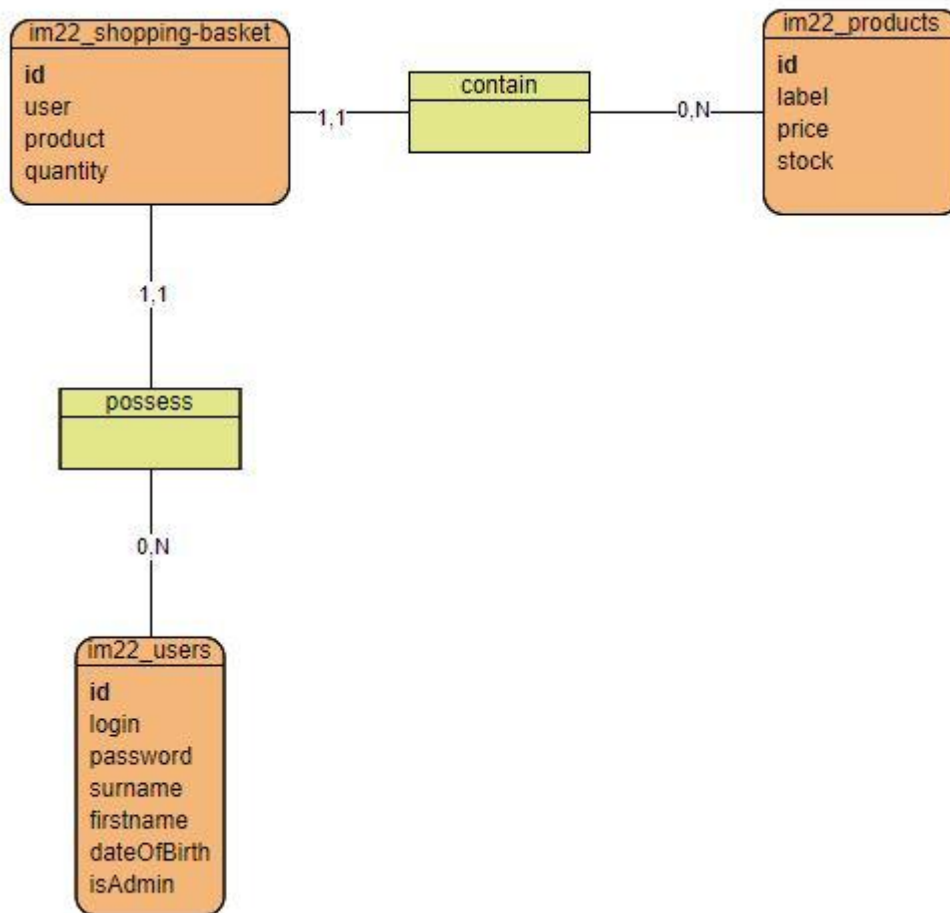
INTRODUCTION	1
SCHEMA (BASE DE DONNEES)	1
ORGANISATIONS DU CODE	3
CRÉATION D'UN SERVICE	4
PARTICULARITÉS	4
CONCLUSION	4



INTRODUCTION

Ce document a pour objet de décrire un site web (créé avec le framework Symfony 5) permettant la vente en lignes d'objet et la gestion de cette dernière par l'intermédiaire d'une base de données. Nous montrerons ici un schéma global de l'organisation du site puis décrirons les différentes composantes que nous avons développées avant de vous expliquer plus précisément la création d'un service avec Symfony, une de ces composantes utiles dans la manipulation du framework.

SCHEMA (BASE DE DONNEES)



Chaque utilisateur peut posséder 0 à n paniers. En effet, un panier correspond à une seule ligne de la table “im22_shopping-basket” et il fait référence à un client ainsi qu’à un produit et sa quantité. En revanche, un panier est bien évidemment personnel et n’appartient qu’à un utilisateur.

De la même manière, un même produit (ex : la pomme A), ne peut pas apparaître dans deux paniers différents. En revanche, il peut y avoir une pomme B dans un autre panier, rattaché à un second client.

ORGANISATIONS DU CODE

Une fois le croquis du site web effectuée, nous avons pu commencer à utiliser le framework Symfony afin de créer le site.

Nous avons utilisé 6 contrôleurs (héritant de la classe *AbstractController*), regroupant des fonctionnalités distinctes. Chacun d'entre eux renvoie vers une série de templates qui affichent les résultats visuels sur le site.

AccueilController.php ne gère que la première page, l'accueil du site.

MenuController.php, grâce à la vue qu'il renvoie, se charge du menu avec les différents liens selon l'identité de l'utilisateur courant. Effectivement le rôle de ce dernier, ses possibilités ne sont pas les mêmes.

PanierController.php se charge de lister les différents produits du panier d'un utilisateur et fait le nécessaire pour ajouter ou supprimer un produit, vider le panier et faire la commande des articles.

ProduitController.php permet l'affichage du magasin et l'ajout d'un produit à ce dernier.

Security.php est essentiel pour l'authentification sur le site web. Il permet aux utilisateurs de se (dé)connecter et nous y avons ajouté la fonction de création d'un compte.

Enfin, **UserController.php** correspond à la gestion utilisateur. Les administrateurs peuvent accéder à la liste des utilisateurs et leurs informations et en supprimer. L'édition de son profil est réalisable par chacun. Quant à l'ajout d'administrateurs, il y a aussi la fonction correspondante, vérifiant alors que l'utilisateur est super-administrateur.

D'autre part, nous avons utilisé trois formulaires symfony, réalisant l'ajout d'un utilisateur (appelé pour l'inscription, l'édition, et l'ajout d'un administrateur), l'ajout d'un produit au magasin ainsi que la connexion d'un client.

Tout contrôleur appelle une vue, disponible dans le répertoire templates, celle-ci héritant de **site.html.twig**, qui hérite elle-même **base.html.twig**.

Ces trois niveaux sont par ailleurs reliés à des fichiers css pour modifier l'esthétique des pages de manière générale, ou spécifique comme nous l'avons effectué sur l'accueil.

Aussi, nous avons créé un service, calculant la somme des entiers d'un tableau et que nous avons affiché sur cette page.

CRÉATION D'UN SERVICE

Pour se faire, nous concevons tout d'abord un nouveau répertoire "Service" qui répertorie l'ensemble des services instanciés. On y instaure ensuite une nouvelle classe php, qui contiendra une ou plusieurs fonctions, utilisable(s) n'importe où dans le code.

Écrivons alors dans le fichier où nous l'utilisons : `"use App\Service\[NOM_CLASSE]"` et nous pouvons par la suite nous servir des fonctions à l'intérieur.

PARTICULARITÉS

Dans notre code, nous aurions pu regrouper certaines parties au sein de mêmes fonctions et ainsi éviter les duplications.

En outre, nous pensons répondre à l'ensemble des spécifications demandées dans le cahier des charges.

Cependant, la réalisation de ces dernières n'a pas été effectuée dans l'ordre recommandé et nous avons préféré réaliser l'authentification dans les débuts de la création de notre site web.

CONCLUSION

Pour conclure, nous avons su, en binôme, travailler de manière régulière afin de répondre aux attentes exigées par l'élaboration du projet et nous avons apprécié la découverte du framework Symfony et des nombreuses possibilités qu'il offre.

Simon KORTSMIT : [Lenspawk \(Github\)](#)

Guillaume PORRO : [BioXtech \(Github\)](#)