

# TP Symfony

## Objectifs

- Être capable de créer un projet Symfony et de le versionner.
- Savoir manipuler les routes et les contrôleurs.
- Savoir créer et exploiter les entités.
- Savoir créer des formulaires et gérer les données utilisateurs avec les validateurs.
- Maîtriser Doctrine avec le mapping, les migrations et le manager.
- Être à l'aise avec Twig.
- Écrire dans les logs de Symfony et envoyer des mails avec SwiftMailer.

## Modalités

Ce projet est à réaliser les journées du 27/08, 28/08 et 29/08.

Vous travaillerez seul ou par groupe de 2 maximum.

Vous devrez versionner votre projet avec git et le rendre disponible sur Github pour le formateur. Vous ferez un maximum de commits avec chaque étape de vos travaux.

Idéalement, vous pouvez travailler avec les Pull Requests sur Github mais ce n'est pas une obligation (Création de branches et *merge* dans la branche *master*).

## Réalisation

Nous souhaitons créer un mini site e-Commerce. L'intégrateur de l'agence vous a livré les maquettes au format HTML / CSS validées par le client. A vous de vous organiser en fonction de ces maquettes. Voici quelques tips pour vous aider dans la réalisation du projet.

Commencez par définir les routes de votre application. En fonction de ces routes, vous allez créer les contrôleurs associés. Par exemple :

- / -> HomeController::index
- /products -> ProductController::list
- /products/iphone-xs -> ProductController::show(\$slug)
- /category/smartphone -> CategoryController::show(\$slug)
- /contact -> ContactController::index
- /cart -> CartController::index

N'oubliez pas les routes du CRUD qui vont correspondre au BO de l'administrateur. Ne passez pas trop de temps là-dessus :

- /admin/products/new → ProductController::new
- /admin/products/edit/1 → ProductController::edit(\$id)
- /admin/products/delete/1 → ProductController::delete(\$id) (uniquement en POST ou DELETE)

Il est évident que chaque contrôleur devra renvoyer une réponse correcte ainsi que la vue Twig associée.

Nous aurons besoin d'une entité métier *Product*. Cette entité contiendra les attributs suivants :

- Identifiant unique
- Nom du produit
- Slug du produit
- Description du produit
- Prix du produit
- Coup de cœur (Booléen)
- Date de création du produit

Le nom du produit devra contenir au moins 3 caractères et la description au moins 10 caractères. Le prix du produit devra être un entier numérique compris entre 99 et X. Le slug du produit sera généré automatiquement à partir du nom du produit. Vous pouvez utiliser une bibliothèque comme [Slugify](#).

Vous vous aiderez de *fixtures* pour remplir la base de données une fois l'entité créée : <https://symfony.com/doc/master/bundles/DoctrineFixturesBundle/index.html>

Vous êtes libre dans la création de produits.

On aura également besoin d'une entité *Category*.

Pour les contrôleurs :

- La page d'accueil affichera 4 produits aléatoires de la base de données, les 4 meilleurs produits (Coup de coeur) et les 4 derniers produits (date de création).
- La page */products* affichera tous les produits dans la base de données.
- La page */products/iphone-xs* affichera le produit iPhone XS de la base de données.
- La page */category/smartphone* affichera tous les produits de la catégorie smartphone.
- La page */contact* affichera le formulaire. Les données du formulaire devront être envoyées par mail à l'administrateur du site (<https://symfony.com/doc/current/email.html>).
- La page */admin/new* affichera un formulaire permettant d'ajouter un produit dans la base de données, vous créerez le formulaire dans les règles de l'art en séparant la création de celui-ci dans une classe externe (Un *FormType*). L'entité devra être validée.
- La page */admin/products/edit/1* permettra de modifier le produit dont l'*id* est située dans l'URL. Vous utiliserez le même formulaire.
- La page */admin/products/delete/1* permettra de supprimer un produit. Cette page ne sera accessible que par la méthode POST de la requête HTTP. On devra donc passer par un formulaire pour supprimer le produit et pas par un simple lien.

## BONUS

A chaque création / modification / suppression d'un produit, on ajoutera cette action dans les logs de Symfony. Cela permettra de surveiller l'utilisation de l'application. Au cas où un client supprimerait un produit par accident.

A chaque demande de contact, en plus d'envoyer un mail, on stockera les données en base de données pour que le client puisse retrouver l'historique des demandes. On créera une entité *Contact*.

On veut également réaliser un système de panier. Le bouton « Ajout au panier » d'une fiche produit permettra d'ajouter le produit en session. Sur la page */cart*, on affichera la liste des produits en session. On devra pouvoir vider le panier ou supprimer un produit de la session.

On aimerait ajouter un système d'avis sur les fiches produits. On aura donc besoin d'une entité *Review* et d'un formulaire pour pouvoir ajouter un avis avec un commentaire et une note.

On peut développer un système de recherche au niveau des produits et/ou des catégories.

Bon courage.