

ToDo & Co

Stack technique

- Php : 8.3
- MySql : 8.0
- Symfony : 6.4
- Doctrine : 2.13
- Php-unit :9.6.20
- Composer : 2.7.7

Si vous souhaitez modifier le projet **ToDo & Co**. Vous pourrez retrouver dans cette documentation technique pour modifier les fichiers ainsi que la logique utilisé pour le système d'authentification et ou vont se trouver les utilisateurs.

Les fichiers que vous pourrez modifier :

Les fichiers les plus important à la modification du projet se trouveront dans le dossier « /src » pour la partie **back end** et le fichier « /templates » pour la partie **front end**.

Vous pourrez trouver plusieurs sous dossier.

Back end :

- Dans le dossier « /Controller » vous allez trouver les fichiers qui vont faire la passerelle entre le **back end** et le **front end**, ils vont contrôler chaque demande entre le client et le serveur.
- Dans le dossier « /Entity » vous allez trouver chaque représentation d'un objet qui seront intégré dans la base de données. Si vous souhaitez ajouter une entité vous pourrez utiliser la commande : **php bin/console make:entity**. Et trouver la documentation sur [Symfony.com](https://symfony.com/doc/current/reference/commands_reference/make_entity.html)
- Dans le dossier « /Form » vous allez trouver chaque formulaire utilisé dans l'application, si vous souhaitez en ajouter un vous pourrez utiliser la commande : **php bin/console make:form**. Et trouver la documentation sur [Symfony.com](https://symfony.com/doc/current/reference/commands_reference/make_form.html)

- Dans le dossier « /Repository » vous pourrez trouver les fichiers liés à une entité qui permettra de faire la liaison entre la base de données et l'application. Vous pourrez ajouter les fonctions spécifiques que vous voulez dans ces fichiers.
- Pour ne pas déranger la logique métier du contrôleur, vous pourrez créer un dossier « /Service » dans le dossier « src », pour créer les fonctions spécifiques que vous voulez

Front end :

- Dans le dossier « /template » vous allez trouver le fichier « base.html.twig » qui l'affichage de base de l'application. C'est-à-dire que chaque fichier dans les sous-dossiers utilisera ce fichier.
- Chaque sous-dossier représente l'affichage d'un contrôleur, ensuite pour savoir quel affichage choisir, tout se gère dans les contrôleurs.

Authentification :

Pour l'authentification, j'ai utilisé ce qui est recommandé par Symfony et j'ai utilisé le package qui est conseillé dans leur documentation. C'est « Security », vous pouvez retrouver des outils de sécurité liés à http, comme les cookies de session sécurisé et les protection CSRF.

Dans notre projet, vous pourrez trouver un fichier .yaml a ce chemin « /config/packages/security.yaml ». Ce fichier est un fichier de configuration pour le package Security.

Ensuite vous pourrez trouver les différents fichiers pour utiliser Security, dans « /src/Controller/SecurityController.php », « /src/Repository/UserRepository.php » et « /template/security ».

Pour ce qui est du fonctionnement du package, lorsque l'utilisateur va renseigner ses informations de connexion et qu'il aura submit le formulaire. Ses informations seront comparées avec les informations qu'il y a dans base de données. En ce qui concerne le mot de passe présent dans la base de données il est hashé donc sécurisé pour éviter qu'on puisse voir directement le mot de passe de chaque utilisateur.

Les utilisateurs :

Les utilisateurs qui vont être créés, vous pourrez les retrouver en base de données et aussi directement dans l'application si vous possédez un compte administrateur.

Dans cette application, vous pourrez retrouver deux types d'utilisateurs. Vous avez un rôle tout simple qui est le « `ROLE_USER` », il pourra utiliser toute l'application. Ensuite vous avez le « `ROLE_ADMIN` » qui pourra administrer les utilisateurs et les tâches non attribuées à un utilisateur.

Le plus :

Pour conserver une bonne qualité de code, pour respecter les PSR, j'ai pu mettre en place deux outils qui pourront être utilisés avant de commit vos modifications sur votre outil de versioning. J'ai mis en place PhpStan et PhpCsFixer.

PHPStan est un analyseur statique pour PHP qui vous permettra de capturer les erreurs avant qu'elles ne se produisent.

Php-cs-fixer est un outil qui permet d'automatiquement contrôler le style du code.