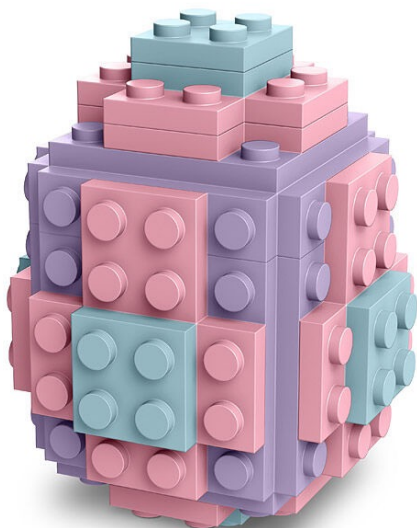


DOSSIER-PROJET

Nom de naissance ▶ DENAËS
Prénom ▶ Pierre-Jacques
Adresse ▶ 17 avenue de la Fade
13600 La Ciotat

Titre professionnel visé

Développeur-se Web et Web Mobile
Titre RNCP - NIVEAU III



<title>

**#MA_PREMIÈRE
AGENCE_WEB.**

</title>

Sommaire

Résumé de projet.....	5
Introduction.....	6
Liste des compétences couvertes par le projet.....	8
Le cahier des charges.....	10
_Club-Convergence.....	11
1. Présentation de l'entreprise.....	11
2. Présentation du projet.....	11
3. Prestations attendues.....	13
_Lutte contre les discriminations La Ciotat.....	15
1. Présentation de l'entreprise.....	15
2. Présentation du projet.....	15
3. Prestations attendues.....	16
Spécifications techniques du projet.....	18
1. Les serveurs.....	18
_serveur local Apache.....	18
_serveur local MySql.....	19
2. Le langage de programmation.....	20
_PHP 7.2.x minimum.....	20
3. Le versionning.....	20
_Git et Gitlab.....	20

4. Le framework.....	21
_Symfony 5.....	21
5. Un gestionnaire de dépendance.....	24
_Composer.....	24
6. Les bundles Symfony.....	24
_WebpackEncoreBundle.....	25
_EasyAdminBundle.....	26
_VichUploaderBundle.....	27
7. Un gestionnaire de package.....	27
_Yarn.....	27
8. Les Librairies JavaScript.....	27
_jQuery-ui.....	28
_jsSocials.....	28
_vanilla.tilt.....	28
_Calendar.....	29
Réalisations et extraits de code.....	30
1. Mixin et variable Scss.....	30
2. Animation menu avec Javascript et Scss.....	32
3. Configuration YAML de l'interface EasyAdmin.....	34
4. Display de compte utilisateur, implémentation de composant Symfony.....	36
5. Display d'affichage d'événement via Twig.....	37
Présentation d'un jeu d'essai.....	39
Veille sur les vulnérabilités de sécurité liées au projet.....	42
1. les attaques par injections SQL (SQLi) et Symfony.....	42
2. la sauvegarde de la base de données.....	

Description d'une situation de recherche.....	45
Traduction d'un extrait de site anglophone.....	47
Conclusion.....	49
Bibliographie.....	50

Résumé de projet

.....#Ma_première_agence_Web.....

Lors de ce qui devait être, ma période de stage en entreprise, six apprenants et moi-même nous sommes retrouvés dans une situation imprévue :

_Le nombre d'entreprises disponible pour l'accueil de stagiaires était insuffisant à cette période.

Un plan B, nous a été trouvé, par notre organisme de formation, consistant à nous répartir en deux groupes pour répondre aux besoins réels émanant de deux structures distinctes :

_Plateforme de lutte contre la Discrimination (POC demandé par un agent de la MPM).

_Club-Convergences (association productrice d'événements musicaux)

Fort d'une expérience plus importante et d'un bon début de formation, j'ai proposé d'organiser le travail des apprenants en une seule entité d'agence Web afin de répondre aux deux projets de manière collective - l'objectif étant de nous mettre en condition réelle.

Une fois l'étape d'agencement (apport de matériel, organisation de l'espace), terminée, le travail a démarré.

Elaboré en un mois « sous » méthode agile Scrum - un rendez-vous client préalable à permis de définir et prioriser les users stories, puis de planifier les sprints sur la base du backlog ainsi défini. Un maquettage des applications a été effectué sur quatre jours et présenté aux clients pour validation. L'architecture des sites que nous avons développés s'apparente à celle des CMS. Nous les avons construits à l'aide du framework Symfony 5. Une conception des bases de données a été pensée dans un premier temps sur papier avec la réalisation du schéma relationnel (mcd mld). Dans un second temps une matérialisation concrète de ces bases a été réalisée avec MySQL WorkBench qui a servi de modèle à la création de nos applications. Un choix d'outils a été fait tant en front-end qu'en back-end comme l'installation d'un bundle d'administration « EasyAdmin » qui crée l'interface utilisateur BackOffice de nos sites.

Avec l'aide de notre formateur, présent un jour par semaine, de la volonté et beaucoup d'heures supplémentaires, nous avons pu répondre au challenge proposé.

Une première version des sites a été proposée, une réponse aux clients a été fournie, « l'équipe de l'agence » est sortie renforcée de cette expérience.

Introduction

En septembre 2019 pour concrétiser une reconversion professionnelle engagée depuis plusieurs mois, j'ai démarré une formation « Développeur Web et Mobile » avec l'organisme Utoplab Factory, basé à La Ciotat.

À la suite à un parcours autodidacte en photographie durant dix ans, je me suis intéressé dans un premier temps au design des sites Web. En effet l'utilisation d'outils que je maîtrisais déjà dans le cadre de mon travail photographique m'ont permis d'entrevoir une possibilité d'études dans ce domaine.

Après plusieurs recherches accompagnées par le PLIE, j'ai pu définir plus précisément mes aptitudes et envies. J'ai réalisé différentes certifications liées à la conception du Web. Celles-ci m'ont permis de découvrir un autre univers que celui du traitement d'image et du design grâce à l'initiation au code et aux langages de programmation. Elles m'ont encouragé à aller plus loin et m'ont ouvert des perceptions différentes en m'offrant des notions de conceptualisation. J'ai alors envisagé le choix d'une formation dans le domaine du Web Développement. Cette discipline répondait mieux à mes attentes en me permettant d'être présent à tous les niveaux de la création d'un site Web ou d'une application.

La formation proposée par Utoplab s'est déroulée sur 700 heures de Septembre 2019 à Mai 2020. Nous avons commencé à 22 apprenants. Les deux premières semaines dites de piscine ou prairie, ont permis d'établir le niveau de chacun. A la suite de celles-ci nous n'étions plus que 19. Plusieurs expériences et mises en situation nous ont été proposées lors de cette période afin de définir les objectifs et les différentes spécialités liées au métier visé. L'un des principes fondamentaux était de nous rendre autonome le plus possible dans nos recherches et apprentissage. Les formateurs nous ont transmis des méthodes afin de dégager et synthétiser les informations dites fonctionnalités, relatives à la conceptualisation de site internet.

La formation prévoyait une période de stage en entreprise de 140 heures de février à mars 2020. Pour ma période de stage, je me suis retrouvé hébergé sur notre lieu de formation, avec six autres apprenants, sans entreprise pour effectuer celle-ci. Notre organisme, aidé de l'un de nos formateurs a recueilli deux contrats pour la réalisation d'un site internet

ainsi que d'un « Proof of concept ». L'objectif étant, pour nous, de nous mettre en conditions réelles.

J'ai alors entrepris, la mise en place d'une agence Web Fictive a laquelle ont adhéré les autres étudiants pour permettre de répondre à deux besoins réels émanants de deux structures distinctes:

_Plateforme de lutte contre les discriminations qui oeuvre pour la libération de la parole.

_Club Convergence, association productrice d'événements musicaux.

Ce travail a été élaboré en un mois « sous » méthode agile Scrum - un rendez-vous client préalable a permis de définir et prioriser les users stories, puis de planifier les sprint sur la base du backlog ainsi défini.

Les principales étapes suivantes se sont succédées :

_Elaboration des cahiers des charges en adéquation avec les besoins clients.

_Choix du stack technique sur recommandation de notre formateur pour nous permettre de respecter la contrainte de temps.

_Versionning du projet sur Gitlab pour la sauvegarde de toutes les étapes et le travail en commun.

_Recherche et sélection d'outils pour la réalisation des besoins spécifiques des clients.

Je vous présente ci-après le résultat de ce travail personnel mais aussi collaboratif.

Liste des compétences couvertes par le projet

Toutes les compétences ci-dessous sont représentées à travers le dossier projet « #Ma_première_agence_web ». Ce dossier regroupe deux projets sur lesquels j'ai travaillé pendant ma période de stage. Plusieurs points des activités types 1 et 2 sont validés et se décrivent comme suit:

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.

Compétences:

1. Maquetter une application.

La réalisation des maquettes a été effectuée avec Balsamic pour le Wireframe ainsi que Photoshop pour la réalisation d'une première maquette au pixel près.

2. Réaliser une interface utilisateur web statique et adaptable.

Les deux projets utilisent le framework Bootstrap dans leur conception, réputé pour sa mise en place simple du côté responsive de nos applications. Afin d'affiner la flexibilité de nos sites sur tout support (pc, tablette, mobile), une liste de media queries côté Css est réalisée.

3. Développer une interface utilisateur web dynamique.

La mise en place d'un calendrier Javascript ainsi que d'un annuaire effectuant un appel Ajax via JQuery aux bases de données des projets est élaborée.

4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.

Une création de gestion publication d'événements à travers un backOffice est construite à l'aide du bundle EasyAdmin et du moteur de template php « Twig ».

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Compétences:

1. Créer une base de données

Les deux projets ont nécessité le développement et la création de base de données avec schéma relationnel SGBRD, d'abord conceptualisées avec MySQLWorkBench puis enregistrées dans le framework Symfony par la création d'entity.

2. Développer les composants d'accès aux données

Sur les deux projets le déploiement du bundle EasyAdmin via Symfony a permis de créer une interface complète gérant le CRUD sur la totalité de nos tables ou entités présentes en base de données.

3. Développer la partie back-end d'une application Web ou Web Mobile.

La partie de développement du backEnd repose sur le fonctionnement et les composants du framework Symfony.

4. Elaborer et mettre en oeuvre des composants dans une application de gestion de contenu ou e-commerce.

Développement d'un composant type article (événements) avec fonctionnalités d'éditations, d'affichages et de partages.

Le cahier des charges

Le cahier des charges est un document qui permet de comprendre et d'expliquer un projet dans son ensemble avec toutes les contraintes, les besoins, les objectifs qui y sont liés.

A la suite du premier rendez-vous « client », nous avons rédigé les cahiers des charges des sites Internet « Club-Convergence » et « Lutte contre les discriminations La Ciotat ». Nous avons amené nos clients à affiner leurs besoins afin de dégager les fonctionnalités prioritaires, dites critiques. L'objectif pour nous étant de pouvoir fournir un premier site fonctionnel dans le délai de trois semaines. Chaque cahier des charges suivra les étapes suivantes:

1. Présentation de l'entreprise.

2. Présentation du projet:

- 2.1 Objectifs du site.
- 2.2 A qui s'adresse le site.
- 2.3 Contenus sur le site.
- 2.4 Arborescence du site.

3. Prestations attendues:

- 3.1 Charte graphique.
- 3.2 Création et récupération de contenus et charte éditoriale
- 3.3 Fonctionnalités.
- 3.4 Dépôt du nom de domaine et adresse mail
- 3.5 Hébergement
- 3.6 Référencement.

CLUB-CONVERGENCE.

1. Présentation de l'entreprise.

Club-Convergence est un lieu unique à La Ciotat. A la fois salle de spectacle et association productrice d'événements musicaux. Située en coeur de ville, elle organise régulièrement des concerts de tout type de musiques. Club-Convergence est aussi un endroit convivial emprunt aux rencontres humaines et artistiques. On y trouve aussi une sélection de bières artisanales ou locales à déguster sur place.

L'objectif principal de l'association est de pouvoir assurer sa visibilité sur le Web. Pour le moment aucun site internet n'existe.

2. Présentation du projet.

2.1 Objectifs du site.

Le site doit pouvoir servir d'outil promotionnel pour l'association et la représenter dans son entité numérique. Celle-ci doit pouvoir communiquer sur les différents événements qu'elle produit à travers une vitrine web de ses activités. Les utilisateurs, doivent pouvoir contacter l'association et se renseigner sur l'agenda des activités à venir. Le site doit faciliter le contact de nouveaux prospects (artiste, mécène, donateur).

2.2 A qui s'adresse le site.

Le site s'adresse en premier lieu au public, aux artistes et aux adhérents. Aucune traduction du site n'est prévue, il sera rédigé en français en évitant les usages et abréviations anglophones. L'association défendant ses origines préférera l'usage du provençal qui est une des variétés de la langue Occitane.

2.3 Contenus sur le site.

Un texte de présentation de l'association est fourni par échange email. Ce texte décrit de manière brève mais complète le lieu Club-Convergence.

Des photos de textures, de rouille, de brique, de béton ciré, de parchemin ou encore de tapis oriental sont fournies au format « .jpeg ».

Les logos des partenaires et mécènes de l'association sont fournis au format « .jpeg ».

Le logo de l'association est fourni au format « .png » et il est de bonne qualité.

Une ébauche de site précédemment tentée est fournie au format « .page ».

L'ensemble des photos fournies par le client ne sont pas utilisables. Les photos sont déjà en compression avancée et ne sont pas manipulables en vue de la création d'un design. Néanmoins, les photos de textures indiquent, une idée générale de l'ambiance. Pour le moment aucun choix réel n'est fait ni tranché. Le site se composera de quatre pages coté utilisateur ainsi que d'une partie BackOffice pour l'administration.

2.4 Arborescence du site.

_Page Accueil ou « Landing Page ».

- inscription News Letter.
- calendrier des événements et tarification.
- présentation de l'événement en cours.
- bandeau partenaires.

_Page d'événements.

- calendrier, planning événements.
- présentation des événements en cours, passés et à venir.

_Page d'informations.

- formulaire de contact.
- localisation.
- bandeau partenaires.

_Page partenaires.

-présentation du caractère bénévole de l'association et de son mode de financement.

-promotion des labels et logos partenaires.

3. Prestations attendues.

3.1 Charte graphique.

Une maquette réalisée précédemment par un autre prestataire est fournie par le client. Cette maquette réalisée avec un logiciel de traitement de texte n'est pas utilisable dans le contexte actuel.

Le logo de l'association existe et est fourni. Une charte de couleur mêlant l'or et le cuivre et des tons noirs sont demandés par le client.

Une recherche d'éléments divers, de design qualitatif, est obligatoire par le prestataire, le contenu fourni n'étant pas utilisable en l'état.

Une police d'écriture est fournie par le client, « NYALA ».

Aucune exigence précise de design n'est imposée, seule la présence, à des endroits précis du logo de l'association et ceux de ses partenaires est requise.

Deux maquettes seront à la charge du prestataire.

3.2 Création et récupération de contenus

La rédaction des textes et informations liés au contenu éditorial est fourni par l'association.

Une recherche d'images et textures sera effectué par le prestataire pour construire le futur design du site.

Le site aura un accès à une interface d'administration, un backoffice pour un à deux administrateurs sera prévu. Le backOffice devra permettre l'édition de contenu, type événementiel, programmation musicale, de consulter les demandes de contact et gérer une photothèque pour un slide.

3.3 Fonctionnalités

_Inscription à la News Letter.

_Gestion et création d'événements avec calendrier.

_Formulaire de contact.

- _Portfolio d'image (slide) alimenté par banque d'images
- _BackOffice d'administration du site.

Un événement présentera, un titre, une photo, une description, une date, un lieu, une option de partage sur les réseaux sociaux et une fonction d'édition permettant aux administrateurs d'activer ou non la publication de celui-ci.

3.4 Dépôt nom de domaine et adresse mail.

Le client dispose déjà d'un nom de domaine qu'il souhaite utiliser.

3.5 Hébergement.

Le client prend en charge l'hébergement et le choix de l'hébergeur de son site.

3.6 Référencement.

Pour ce qui est du référencement, l'association Club-Convergence, n'a pas d'objectif ni de vue précise de ses besoins. Un site respectant les bonnes pratiques du développement web apportera dans un premier temps, un référencement naturel.

L'association n'a pas de budget ni les moyens pour se faire accompagner dans son développement numérique. Le projet est un premier site pour l'association.

LUTTE CONTRE LES DISCRIMINATIONS LA CIOTAT.

1. Présentation de l'entreprise.

La demande de projet émane de Sophie Delage, Cheffe de service à la Maison de la cohésion sociale, sur le territoire de la ville de La Ciotat. Sophie Delage fait partie de la MPM et de l'équipe opérationnelle de la politique de la ville.

Le projet sera un Proof of Concept, une plateforme permettant de centraliser les informations des associations luttant contre la discrimination en territoire Paca, essentiellement destiné en premier lieu aux associations de la ville de La Ciotat.

2. Présentation du projet.

2.1 Objectifs du site.

Assurer l'information ainsi qu'une présence web à destination de la Ciotat sur les sujets de discrimination. Donner un outil de consultation, communication et de recherche aux différentes associations du territoire. Regrouper tous les partenaires et réseaux de la lutte pour avoir une vue globale de l'action en Paca. Le site doit permettre une communication sur les événements créés par la plateforme. Il sera possible d'alerter en cas de discrimination ce qui donnera la possibilité à la plateforme de pouvoir diriger les utilisateurs vers les associations en correspondance avec le problème. Centraliser les demandes d'aides et de contacts des utilisateurs.

2.2 A qui s'adresse le site.

Le site s'adresse en premier lieu, au réseau associatif de la région Paca et plus particulièrement de la ville de La Ciotat. Le public est la seconde cible du site. Pour le moment le projet étant un proof of concept, aucune traduction du site n'est envisagée.

2.3 Contenus sur le site.

Une liste des critères de discrimination est fournie, il y en a 25 au total, au format « .txt ».

Un flyer d'événement organisé par la plateforme au format « .jpeg » est transmis. La liste de toutes les associations, coordonnées et référents est fournie au format « .csv ».

Un descriptif de la mission de la plateforme ainsi que sa présentation doivent être fournis par le client.

Le site se composera de cinq pages. Une page d'accueil, une page événementiel, une page information annuaire association, une page de formulaire d'alerte, une page de formulaire de contact. Un backOffice doit être prévu.

2.4 Arborescence du site.

_Page d'accueil

- présentation mission de la plateforme.
- mise en avant événement en cours.
- bandeau contact et réseaux sociaux.

_Page événementiel

- calendrier, planning événements.
- présentation événements passés, en cours ou à venir.

_Page d'informations

- Recherche annuaires associations et affichage.
- bandeau contacts et réseaux sociaux.

_Page Alerter

- formulaire d'alerte discrimination.

_Page contact

- _formulaire de contact.

3. Prestations attendues.

3.1 Charte graphique.

Le site est un « Proof of concept », une démonstration de faisabilité basée sur quelques idées et besoins du client. Aucune charte graphique n'est définie, elle sera choisie sur la base de conception du flyer d'événement. Aucune police d'écriture n'est requise ou imposée. Un design simple est recommandé. Le prestataire devra être force de proposition.

3.2 Création et récupération de contenus.

La rédaction des textes et informations liées au contenu du site sera fournie par la plateforme. Le client n'ayant que peu de temps à attribuer au contenu général et le design du site étant sommaire, quelques photos, images seront à la charge du prestataire. Le site disposera d'une interface d'administration avec différents rôles de gestion utilisateur. Un Super_Admin devra pouvoir pratiquer le CRUD sur la totalité des informations inscrites en base de données. Le backOffice permettra l'édition ou la lecture suivant les rôles de contenu événementiel ou d'informations associatives.

3.3 Fonctionnalités

- _Création BackOffice avec gestion de contenu.

- _Annuaire interne et externe recherche associations.

- _Gestion des Rôles d'administration du BackOffice, activation désactivation d'utilisateur.

- _Publication d'événements, activation ou désactivation de publication.

- _Formulaire d'alerte discrimination avec possibilité d'édition statistique en interne.

- _Formulaire de contact.

Un événement présentera, un titre, une image, une date, une description, un lieu, une option de partage sur les réseaux sociaux.

Les cahiers des charges proposés ont été validé par nos clients. Compte tenu des délais très courts d'environ trois semaines, de l'absence de budget et d'un niveau très hétérogène dans notre équipe de production, un stack technique a été retenu. Avec l'aide de notre formateur, présent un jour par semaine, de la volonté et beaucoup d'heures supplémentaires, nous avons pu répondre aux objectifs fixés.

Spécifications techniques du projet

Suite au recueil des informations clients et à la réalisation des cahiers des charges, nous avons pu relever des similitudes dans les deux projets en termes de fonctionnalités.

L'un des projets, « Plate-forme de lutte contre les discriminations la Ciotat », était plus complet dans son développement. Suivant le choix du stack technique, il était possible de concevoir le site le plus avancé pour ensuite se resservir de la structure pour l'autre.

La validation de notre stack technique s'est donc réalisée à partir de plusieurs critères.

L'apprentissage d'un Framework PHP était l'un des points de validation de notre formation.

Nous avons développé sur serveur local, utilisé le langage PHP, fait appel à des gestionnaires de dépendances ou de package, installé des « bundles » et inclus des « library ».

Nos projets ont été versionnés pour permettre une sauvegarde avec traçabilité grâce à un journal de « log » mais aussi le développement commun sur un système de branche (branch) évitant d'altérer les projets.

Je me propose ici de vous faire la liste ainsi qu'une présentation introductive de notre stack technique.

1) Les serveurs:

_ Le serveur local Apache.

Le serveur apache est un logiciel libre dit « open source ». C'est un serveur http dont la première version remonte à 1995. Ce n'est qu'à partir de sa version 2 que le serveur sera largement utilisé à travers le monde. Aujourd'hui encore Apache Server alimente environ 46 % des sites Web de la planète.

Le serveur permet de servir du contenu sur le Web ou de simuler en local en se comportant comme un virtual host. Le virtual host ou hôte virtuel est capable de gérer plusieurs sites Web sur la même machine. Le travail du serveur Web est de fournir les fichiers stockés demandés lorsque l'on rentre un nom de domaine dans la barre d'adresse

de notre navigateur Web. Le serveur Web traite les fichiers écrits en différents langages de programmation telle que PHP, Python, Java, JavaScript et autre. Il les transforme ensuite en fichiers HTML statiques et diffuse ces derniers dans le navigateur Web des utilisateurs.

Le serveur Web apache n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur. C'est l'outil responsable de la communication serveur client. Il établit une connexion entre serveurs et navigateurs Web client. Le navigateur (Google Chrome, Firefox, Opéra, Safari, etc) envoie une requête au serveur qui renvoie une réponse avec les fichiers demandés (texte, image, etc).

Apache est responsable de la communication fluide et sécurisée entre serveurs et clients. Un fichier « .htaccess » permet une personnalisation et configuration complète du serveur. Le plus grand défi d'un serveur Web est de servir simultanément plusieurs et différents clients. Le serveur Apache après plus de 20 ans d'existence reste encore le serveur Web le plus utilisé dans le monde, c'est la référence à connaître et c'est en ce sens que notre choix se fera.

Le serveur local MySQL.

Le serveur MySQL est un système de gestion de base de données relationnelles. Il est distribué sous licence propriétaire mais aussi GPL, licence publique générale GNU, il est donc open source. Le serveur MySQL est un des logiciels de base de données les plus utilisés dans le monde. SQL fait référence au langage de requête utilisé (Structured , Query, Langage).

Une base de données n'est qu'une collection structurée de données organisées pour en faciliter l'utilisation et la récupération. Dans une base de données relationnelles nos données sont divisées en plusieurs zones de stockage appelées « table ». Cela permet d'agencer les données différentes et d'éviter la duplication de celle-ci. Par exemple lorsqu'un client passe une commande, si celui-ci a déjà un compte utilisateur, nous n'allons pas réinscrire les informations utilisateurs en base de données à chaque nouvelle commande. Nous pourrions séparer les données de commande de celles des clients et les faire correspondre grâce au schéma relationnel. Nous avons donc une meilleure organisation des données. Les données des tables se lient entre elles grâce un système de « clé » afin de pouvoir les manipuler les combiner dans différentes tables si nécessaire.

Sans serveur local aucune conception n'est concrètement envisageable. Le choix des serveurs Apache et MySQL pour réaliser notre production en local vont dans le sens de notre apprentissage du développement Web. Ces serveurs, étant les seuls utilisés et appris par les apprenants, par confort et gain de temps, notre choix se porte donc sur ces deux références.

2) Le langage de programmation.

_PHP 7.2.x minimum donc 7.3

PHP est un langage de programmation côté BackEnd. Le langage de programmation va communiquer avec le serveur. C'est un langage de programmation libre il est donc open source. Il est multi-plateforme et peut être utilisé autant sur Mac que Linux que Windows. PHP est un langage impératif orienté objet. Il est principalement utilisé pour produire des pages Web dynamique via un serveur http. Les versions PHP 5.6, 7, 7.1 ne sont officiellement plus supportées. D'après la W3Techs (Web Technology Survey) PHP est utilisé par plus de 78,9 % des sites Web qui se servent d'un langage coté serveur. L'un des problèmes actuels est que beaucoup de ces sites utilisent encore des anciennes versions non supportées de PHP. Seul 38,5 % d'entre-eux fonctionnent avec des versions de PHP supportées (7.2 ou plus) ce qui introduit des problèmes de performance et de sécurité. L'utilisation de la version PHP 7.3 étant majoritaire dans le développement d'aujourd'hui, sa communauté étant la plus importante, PHP 7.3 sera notre langage de programmation sur les projets.

3) Le versionnage.

_Git et Gitlab.

Git représente l'outil et Gitlab, la plateforme de versionnage.

D'autres plateformes comme BitBucket, SourceForge, Aws Code Commit , ou encore Cloud Source existent. Le choix s'effectue surtout par préférence d'utilisation. Le versionnage est la création et la gestion de plusieurs versions d'un produit. Le contrôle de version est la pratique consistant à fournir un partage des données et une édition collaboratives entre les utilisateurs. Schématiquement, on passera de la version N à la version N + 1 en

appliquant une modification M. Un logiciel de gestion de versions apporte un suivi des modifications une par une pour fournir la version du fichier voulue. Nous commencerons par créer notre dépôt en nous plaçant dans le dossier de notre projet et en effectuant la commande « git init » nous ajouterons un premier fichier avec la commande « git add (mon_fichier) » et nous le commiterons. Pour ce faire nous utiliserons donc la fonction « commit » ici dans git « git commit -m « mon message » » pour réaliser un nouveau commit à la suite d'un ajout de fichier dans notre dépôt (repository ou repo) ou « git commit -a -m « mon message » » dans le cas où notre modification porte sur un fichier déjà présent sur notre dépôt. Le commit est une sorte d'instantané (photo) de notre dépôt après avoir effectué toute sorte de modification. Celui-ci nous sert de point de contrôle nous donnant la possibilité de ré-évaluer ou restaurer notre dépôt à un état précédant. Le commit sert à enregistrer une nouvelle version d'un ensemble de fichiers sur notre dépôt. Un système de branche qui signifie diverger de la ligne principale de développement et continuer à travailler sans se préoccuper de cette ligne principale, permet de faire travailler plusieurs personnes sur un projet en même temps sans que la ligne principale ne soit altérée par les modifications et ajouts de chacun.

Les branches permettent d'avoir un historique non linéaire et permettent de travailler pendant un moment sur une fonctionnalité particulière sans pour autant "polluer" le flux principal. Elles peuvent servir pour travailler sur des fonctions spéciales mais aussi être utilisées pour garder une bonne organisation avec des branches qui correspondent à nos différents environnements (master, recipe, dev, bugfix...). Cela permet le développement parallèle de plusieurs fonctionnalités volumineuses sans bloquer le travail quotidien sur les autres fonctionnalités et la maintenance d'anciennes versions du logiciel (sur les branches) tout en continuant le développement des futures versions (sur le tronc).

Les branches sont des copies du projet principal avec leurs propres versions complètes des modifications qu'elles ont chacune produites de leur côté. Après avoir terminé, il sera temps de « fusionner » ces branches pour les ramener vers la branche « master », le répertoire principal du projet. Le choix de gitlab s'est fait en rapport à ses valeurs open source, github venant d'être racheté par Microsoft.

4) Le framework.

_Symfony 5.

Nos projets nécessitent une interface de gestion de contenu. Ils devront pouvoir être accessibles par un grand nombre d'utilisateurs et avoir une architecture robuste sécurisée. La solution la plus rapide et sécurisée dans le développement de la partie BackEnd de nos projets, ayant une deadline à trois semaines se trouve dans le choix et l'utilisation d'un framework.

Un framework, est comme son nom l'indique, un cadre de travail. Il est toujours lié à un langage de programmation (PHP, Python, Java, Javascript...). Il apporte de nombreux outils qui permettent aux développeurs de ne pas « réinventer la roue » (expression typique dans le développement web). Les directives et le cadre fournis par le framework permettent une structuration du code qui par conséquent améliorent sa qualité. Son objectif est aussi d'harmoniser les pratiques de développement ce qui en BackEnd est assez pertinent.

Aujourd'hui des frameworks sont recommandés dans le cadre de développement de site web de taille importante, ils favorisent une maintenance plus facile sur le long terme.

Ils peuvent néanmoins être une couche de complexité supplémentaire, prévoir un temps de formation, peut-être nécessaire pour les appréhender.

Il existe différents types de framework:

_Les frameworks maisons, développés en interne par des entreprises sur la base de leur expérience.

_Les frameworks payants, développés par des éditeurs de logiciels.

_Les frameworks OpenSources, développés et maintenus par une communauté.

Dans notre cas c'est ce type de framework « OpenSource » qui nous intéresse. Le caractère OpenSource nous garantit une gratuité framework. L'OpenSource permet à chacun de s'attribuer le code d'un programme, environnement ou autre, d'y contribuer et de le redistribuer à sa guise.

Afin de choisir notre framework, deux indicateurs importants sont à prendre en considération :

_Le taux d'usage, dans le monde du framework, il y a un taux de perte important, beaucoup sont abandonnés et viennent à ne plus être maintenus.

_La communauté, plus celle-ci est importante, plus il y a de forums d'entraide, plus on trouvera de réponses facilement à nos problématiques et plus la maintenance et le développement seront assurés.

Pour finir l'un des objectifs de notre formation était d'apprendre un framework PHP, notre choix se portera donc sur Symfony 5 (dernière version stable). Symfony est un framework français mondialement connu orienté BackEnd. Il est actuellement parmi les frameworks les plus plébiscités par les développeurs d'aujourd'hui. Pour en citer d'autre, Laravel, CodeIgniter, CakePhp sont des concurrents directs. Symfony est particulièrement apprécié pour ses performances et sa simplicité d'utilisation. C'est un ensemble de composants (ou « bibliothèques »), qui facilitent le développement Web, en réduisant de façon considérable, le temps et l'effort requis pour créer des composants génériques. Le framework utilise une méthode de programmation de type « Ruby On Rails » qui suit le modèle de conception « MVC », (modèle, vue, contrôleur) et respecte le principe « DRY » (don't repeat yourself).

Symfony simplifie le développement en lui donnant une structure en pleine conformité avec les règles métiers, à la fois maintenable et évolutive. Toutes les bonnes pratiques de développement regroupées dans Symfony permettent aux développeurs de rendre un code lisible et accessible. La maintenance dans le temps en est facilitée et cela permet à n'importe quel développeur de pouvoir reprendre le projet. Les pratiques sont harmonisées. Symfony tire sa logique de la POO (programmation orientée objet). La POO est un paradigme en informatique employé pour exprimer, la façon dont un système a été conçu et pensé dans ses grandes lignes.

Symfony rend la gestion des bases de données accessible en faisant abstraction des différences entre les moteurs de bases de données et en convertissant les éléments relationnels sous forme de classes orientées objets. Dans Symfony on crée des « Entity » qui représentent des classes PHP. Une « Entity » n'est pas forcément une table dans notre base de donnée, une « Entity » représente une logique métier.

La logique métier est tout le code personnalisé, écrit pour notre application et qui n'est pas spécifique au framework. En informatique, c'est la partie du programme qui détermine comment les données peuvent être créées, stockées et modifiées.

Dans Symfony, une « Entity » est rattachée à un « Repository » qui représente lui un objet.

L'un des autres avantages de Symfony est qu'il intègre, Bootstrap ainsi que jQuery dans son squelette. Bootstrap est aussi un framework ou collection d'outils, pour le développement côté Front-end. Il est écrit en JavaScript, CSS, HTML et Sass. Bootstrap est un framework Css c'est à dire côté design. Il est cross-browser ce qui permet au site de garder une présentation similaire quel que soit le navigateur web utilisé. Il est responsive ce qui facilite grandement le développement à destination de multi support, téléphone, tablette, ordinateur, grand écran. Bootstrap embarque des composants JavaScript, il est totalement open source, il offre un système d'agencement du design sur 12 colonnes. Son harmonisation du design et sa normalisation de présentation peuvent néanmoins devenir lassante.

Symfony étant notre choix pour le développement, celui-ci intégrant Bootstrap nous utiliserons donc les deux pour l'ensemble de nos projets.

5) Un gestionnaire de dépendance.

_Composer.

Depuis la version de PHP 5, l'ajout des objets et des namespaces a rendu le langage modulable et réutilisable. Des milliers de composants et de bibliothèques ont été créés par la forte communauté existante. Ceci a vite généré, la création d'un outil nécessaire pour les gérer.

Composer est un outil multi-plateforme (mac, linux, windows) de gestion de dépendances en PHP. Il est à l'image de « npm » pour NodeJs. Il a été mis en place pour aider la communauté des développeurs PHP et permet de gérer les dépendances entre applications et librairies. Le gros avantage de Composer est qu'en rapport à un gestionnaire de package, Composer assure l'installation mais surtout le suivi et la mise à jour des composants et bibliothèques dont nos projets ont besoin. Utilisant PHP ainsi que Symfony, Composer fait partie de la suite logique, des éléments constituant notre stack. Dans notre

cas Composer a géré tous les « Bundles » qui ont été utiles à la conception et au développement de nos deux projets.

6) Les bundles Symfony

Le bundle est une sorte de plug-in ou module permettant d'implémenter plusieurs fonctionnalités dans nos projets. En PHP il est considéré comme un namespace. Les bundles sont capables de modifier le comportement de notre application.

Un des gros avantages est qu'ils sont open source, totalement réutilisables dans la plus part de nos projets. Sur Symfony il existe des milliers de bundles et même des sites comme KpnBundle qui vous proposent de les télécharger. Encore une fois le principe du bundle est de ne pas réinventer la roue et de nous procurer un gain de temps considérable dans les phases de développement de nos sites. En plus d'être réutilisable les bundles sont flexibles mais surtout sécurisés. Par exemple EasyAdminBundle est un puissant outil d'administration de site sous Symfony. Dans nos projets avons fait appel à lui mais aussi à Webpack-Encore-Bundle ainsi que VichUploaderBundle.

Je vous détaille ci-après les éléments ayant notifié le choix de ces bundles.

Webpack-Encore-Bundle.

Webpack-Encore-Bundle a été le premier bundle installé par notre équipe. Ce bundle est une API (interface de programmation d'application). Il regroupe et gère la fonction de plusieurs modules JavaScript. C'est une bibliothèque Javascript qui va nous permettre le pré-processing Css et Js ainsi que la compilation et minification de ceci. Cette API nous permet de manager notre Css et notre Js. En vue de la customisation de Bootstrap, directement implanté dans Symfony ce bundle est apparu comme une évidence. La configuration Css de Bootstrap est écrite en SASS ou SCSS. Afin de modifier celle-ci nous avons besoin d'un pré-processeur css. De plus l'utilisation d'un pré-processeur Css coté Front-End permet un meilleur agencement et donne une logique au code design ainsi que des fonctionnalités.

Une fois le bundle installé celui-ci est fourni avec un fichier de configuration « webpack.config.js » qui nous servira à activer les différentes fonctionnalités ou

méthodes. Dans notre cas nous activerons le module « sass-loader » pour l'ensemble de nos projets. Une fois la gestion du module pris en charge par notre application, nous avons dû installer l'ensemble du Webpack-Encore ainsi que le module du Sass-loader avec Yarn, un gestionnaire de package JavaScript. Yarn nous servira également à exécuter les commandes liées à la réalisation du pré-processing.

_EasyAdminBundle.

EasyAdminBundle est un bundle mis en avant par Symfony. C'est un bundle d'administration de nos « Entity » créant un BackOffice pour nos sites Web. Ce backOffice est livré avec une gestion du CRUD (Create, Read, Update, Delete), une fonction de recherche, une pagination et des templates responsives, le tout, clé en main !

Une fois le bundle installé, la création d'une première vue du BackOffice ne prendra que quelques secondes. Toute la configuration d'EasyAdmin se réalisera via un fichier « .YAML ». Cette configuration est relativement verbeuse mais considérée comme « human-friendly ». Un des gros atout d'EasyAdmin est qu'il repose sur des concepts standards. Par exemple, pour ajouter une fonctionnalité de type CMS, gestion de média, il suffit de créer un contrôleur et de passer sur du code Symfony. Ce bundle est très simple à mettre en place et il est possible de développer et d'intégrer des fonctionnalités avancées en son sein. EasyAdmin, étant poussé par Symfony et allant exactement dans le sens de nos besoins c'est lui que nous choisirons. De plus ce bundle peut se compléter avec d'autres composants eux aussi sous forme de bundle. VichUploader est l'un des composants que nous avons ajouté au fonctionnement d'EasyAdmin.

_VichUploaderBundle.

VichUploaderBundle est un bundle qui tend à faciliter le téléchargement de fichiers, attachés à nos entités « Entity ». Il permet le nommage et l'enregistrement des fichiers dans un dossier précis de notre application. Il gère la suppression de ces mêmes fichiers du système en cas d'effacement de l'entité rattaché ou encore de la base de données. Dans notre cas VichUploader servira à l'importation de fichier type image (jpeg, png) et document (Pdf) dans l'interface BackOffice d'EasyAdmin. Le bundle ajoute automatiquement, quand cela est possible, une miniature du fichier dans l'affichage par liste des entités « Entity », ce qui permet un meilleur repérage du contenu dans notre interface Back.

Comme nous pouvons le voir, les bundles sont de conceptions diverses. Un bundle peut représenter une simple fonctionnalité comme il peut représenter un ensemble complexe de mise en place et d'interactions. Certains bundles comme le bundle WebpackEncore sont des bibliothèques Front-End regroupant plusieurs modules Javascript qui nécessiteront un gestionnaire de package Js pour assurer leur fonctionnement. Les bundles sont une ressource non négligeable dans le développement Web. Grâce au gain de temps qu'ils nous apportent, il est rapidement possible, de mettre en place des architectures élaborées, robustes et sécurisées.

7) Un gestionnaire de package.

_Yarn.

Yarn est un logiciel Open Source. Il permet la simplification de la gestion de dépendance dans nos projets. Yarn est un gestionnaire de package JavaScript pour NodeJs développé par Facebook et visant à améliorer les lacunes de « npm ». Yarn est compatible avec une grosse quantité d'infrastructures actuelles. Yarn est compatible avec « npm » et « bower » ce qui permet de limiter le nombre de gestionnaires de packages par projet. Pour éviter les problèmes de version de package, Yarn met à jour un fichier « yarn.lock » avec des numéros de versions fixes. Cela permet d'installer, d'une machine à l'autre exactement les bonnes versions de tous nos packages. Un des gros avantages de Yarn est qu'il entrepose les dépendances en cache ce qui permet d'éviter un re-téléchargement de celle-ci à chaque nouveau projet. Yarn nous permettra d'installer et d'exécuter toutes les dépendances de notre projet coté Front-End. Par exemple, pour nos projets Symfony, nous avons installé via composer le bundle Webpack-Encore. Ce bundle intègre et configure Webpack-encore dans nos projets ensuite il faudra installer et exécuter la bibliothèque via le gestionnaire Yarn. En plus d'améliorer « npm » Yarn assure une sécurité absolue des dépendances qu'il gère. Yarn est un choix nécessaire dans notre volonté d'utiliser le format Scss pour la conception du design des deux sites web.

8) Les bibliothèques JavaScript.

Les bibliothèques JavaScript sont une source intarissable de ressources pour les web designers et développeurs front-end. Grâce au langage JS, vous pourrez créer des designs modernes et des fonctionnalités fluides pour nos utilisateurs. Suivant les tendances du webdesign, les bibliothèques JavaScript nous sont utiles pour créer des interfaces from scratch ou améliorer l'expérience utilisateur de nos sites web. Les bibliothèques javascript sont des frameworks Front-End

_jQueryUi.

jQuery UI veut dire, jQuery User Interface. jQuery UI est en licence MIT et est open source. Cette bibliothèque JavaScript offre des méthodes additionnelles qui complètent parfaitement jQuery. jQuery est l'un des composants Front-End de nos applications Symfony. Les ajouts que jQuery UI apporte, sont appliqués à l'interface utilisateur.

C'est une bibliothèque d'assemblage de plugins accessible grâce à un seul fichier JavaScript. jQuery UI propose un panel d'interactions utilisateurs sur les éléments du « DOM » (document object model), une collection de widgets et d'effets visuels ainsi que deux utilitaires. Cette bibliothèque peut autant servir à construire des applications Web hautement interactive ou encore à simplement utiliser des éléments ou fonctions uniques suivant les besoins recherchés.

Dans notre cas une fonction de recherche avec auto-complétion devait être installée sur l'une des vues utilisateurs de l'un de nos projets web. jQuery UI assurant une compatibilité parfaite avec nos applications, il fut un choix rapide exécutant le travail.

_Vanilla-tilt.

Vanilla-tilt est une bibliothèque JavaScript à inclinaison 3D écrite en vanilla (JavaScript pur), c'est un fork ou reprise de Tilt.js (version jQuery). Vanilla-tilt est en licence libre MIT. Cette bibliothèque n'a aucune dépendance et elle est très légère. Sa seule fonction est d'opérer différents types d'inclinaisons 3D sur des boutons ou surfaces de nos sites.

Vanilla-tilt nous a permis d'intégrer quelques effets visuels attractifs de manière simple et rapide.

_JsSocials.

JsSocials est aussi sous licence MIT et est open source. Ce n'est pas vraiment une librairie JavaScript, JsSocials est un plugin jQuery. Il permet facilement de mettre en place un système de partage sur les réseaux sociaux. Il fournit plusieurs thèmes et il est possible d'ajouter tout réseau social encore non pris en charge par le plugin.

Nos deux projets nécessitent une gestion de contenu de type événementielle. Afin de rendre ce contenu destiné au public partageable, nous avons donc pu intégrer ces liens grâce à JsSocials. JsSocial est très léger et ne dispose que de cette seule fonction.

_Caleandar.js

Caleandar est un script de calendrier ultra léger (7,5kb). C'est un petit projet JavaScript sous licence libre MIT que nous avons trouvé sur le gitHub de son développeur « jackducasse ». Ce calendrier est un script objet indépendant de bibliothèques ou de thèmes facultatifs. Il dispose d'un setting simple et allant à l'essentiel. Nos deux projets web, nécessitant tous deux un calendrier dans lequel il soit possible d'instancier des événements, la légèreté et la simplicité de Caleandar, s'est révélée comme un choix efficace pour notre équipe de travail.

Réalisations et extraits de code

Durant la création des projets, je suis passé autant du côté back-end que front-end. J'ai pu coder une partie du design grâce au scss, réaliser une fonction d'animation du menu avec Javascript ou encore effectuer la configuration du fichier YAML de EasyAdmin mais aussi mis en place une fonction display sur la gestion des comptes utilisateurs ainsi que dans celle des événements.

Je vous présente ci-joint une liste de mes réalisations ainsi que les extraits de code les représentant :

- 1. Mixin et variable Scss.**
- 2. Animation menu avec Javascript et Scss.**
- 3. Configuration YAML de l'interface EasyAdmin**
- 4. Display de compte utilisateur(admin) par implémentation de composant Symfony.**
- 5. Display d'affichage d'événement via Twig.**

La première étape dans le développement de nos applications, a été d'intégrer un pré-processeur Css, ici Sass(Scss) dans notre framework Symfony.

1.Mixin et variable Scss.

Quand on commence à utiliser un pré-processeur Css, il devient vite difficile de revenir en arrière. Le Scss permet d'organiser et d'écrire plus vite du Css. Il se rapproche plus d'un langage de programmation qui nous donne la possibilité d'éditer des variables, fonctions, mixin etc. Grâce à des « @include » le découpage et la ré-utilisation du Css à différents endroits est simplifié. Le Scss dispose de plusieurs niveaux d'imbrication qui permettent une factorisation du code et ainsi appliquer une méthode « Dry » sur notre code. Dans nos projets, je fais appel à Sass afin de personnaliser BootStrap. En effet les fichiers source de BootStrap sont écrits en Scss et leur import pour modification est possible directement au sein de notre code scss. Voici un extrait de mon fichier « _variable.scss », lui-même inclut (@include) dans un fichier général « app.scss ». Dans

premier temps j'édite des variables de couleur ce qui m'évitera de re-taper un code à chaque utilisation. Je personnalise la variable « \$theme-colors » de bootstrap en lui ajoutant mes nouvelles couleurs. Ensuite je crée les nouvelles classes « btn » en relation avec les nouveaux thèmes. Ensuite sachant que je vais utiliser plusieurs fois un système d'ombre dans le design ainsi que des arrondis, j'écris quelques mixin pour pouvoir y faire appel chaque fois que j'en aurai besoin.

```
/* variables couleurs */

$colorM: hsl(48, 98%, 50%);
$colorBtm: hsl(198, 64%, 41%);
$colorShadow: hsl(0, 0%, 15%);
$colorPurple: hsl(283, 70%, 58%);

/* ajout de nouveaux thèmes couleurs dans Bootstrap */

$theme-colors: (
  "yellowM": $colorM,
  "blueB": $colorBtm
);

/* ajout des nouveaux thèmes à la classe btn- */

@each $key, $color in $theme-colors {
  .btn-#{ $key } {
    background-color: $color;
  }
}

/* Mise en place de mixin */

// Mixin box-shadow
@mixin shadow($val){
  @if $val == general {
    box-shadow: -2px 2px 5px 0px $colorShadow;
  }
  @else if $val == card {
    box-shadow: 15px 15px 7px 0 rgba(0,0,0, 0.4);
  }
  @else if $val == bottom {
    box-shadow: 0 15px 15px -15px $colorShadow;
  }
  @else if $val == top {
    box-shadow: inset 0 15px 10px -15px $colorShadow;
  }
  @else if $val == inset {
    box-shadow: inset 0px -5px 5px 0px rgba(0,0,0,0.51);
  }
}

// Mixin border-radius
@mixin radiusBtmL {
  border-radius: 0 0 0 15px;
}
```

Extrait de code Scss 1

2. Animation du menu avec JavaScript et Scss.

Pour réaliser, l'animation d'ouverture et de fermeture de l'un de nos menus, j'ai écrit en premier la partie code Scss correspondant au design de celui-ci. J'ai créé différentes classes Css pour chaque état, position et même comportement de chaque élément du menu en position - ouvert ou fermé.

```
// menu position fermé
.yellowMenu {
  z-index: 999;
  width: 100vw;
  height: 300px;
  background-color: $colorM;
  position: absolute;
  top: -150px;
  left: 0;
  transform: skewY(17.6deg);
  transition: 0.2s;
  @include shadow(bottom);
  // menu position ouvert
  &-open {
    top: -100px;
    transition: 0.2s;
    transform: skewY(18.0deg);
  }
}
```

Extrait Scss 2: code représentant l'icon burger de mon menu. Un premier état de l'icon en position fermée est défini par une classe Css « .svgBtnMenu » suivi d'une modification de cette état par une seconde classe en position ouverte « .svgBtnMenu-open ». L'icon passe d'un état forme burger (trois ligne superposées) à l'état de croix avec un effet de transition et de changement de couleur.

```
//icon burger position menu fermé
.svgBtnMenu {
  width: 70px;
  height: 70px;
  #top, #middle, #bottom {
    transition: 0.3s;
    stroke: black;
  }
  #top {
    transform: translateY(-80px);
  }
  #bottom {
    transform: translateY(80px);
  }
  //icon burger position menu ouvert
  &-open {
    opacity: 1 !important;
    #top, #middle, #bottom {
      transition: 0.3s;
      stroke: $colorBtm;
    }
    #top {
      transform: translateY(0) rotate(45deg);
      transform-origin: center center;
    }
    #middle {
      opacity: 0;
    }
    #bottom {
      transform: translateY(0) rotate(-45deg);
      transform-origin: center center;
    }
  }
}
```


Extrait Scss 3: code représentant le conteneur du menu en position : fermé (.yellowMenu) suivi de sa position ; ouvert (.yellowMenu-open). Chaque classe représente un état de l'élément. Sa taille, sa position, sa transformation ainsi que la durée de transition sont instruites. Ici l'élément représenté est une div rectangulaire jaune dans un premier état fermée, en position inclinée. Une fois le menu ouvert, le rectangle se positionne horizontalement. Une fois nos différents éléments créés, il ne reste plus qu'à mettre en place une fonction JavaScript pour switcher entre les classes ouvertes et fermées.

Extrait JavaScript 1: on commence par déclarer une variable représentant une sélection de l'élément icon burger de notre menu. On applique une fonction onclick sur notre variable dans laquelle on exécute une méthode toggle pour chaque élément de notre menu à modifier de position fermée à ouvert.

```
// fonction de switch de classe ouverte fermée
let openMenu = document.querySelector("#openMenu");
openMenu.onclick = function () {
  const burger = document.querySelector(".svgBtnMenu");
  burger.classList.toggle("svgBtnMenu-open");
  const menuB = document.querySelector(".yellowMenu");
  menuB.classList.toggle("yellowMenu-open");
  const navigate = document.querySelector(".navigate");
  navigate.classList.toggle("navigate-open");
  const iconbtn = document.querySelector(".icon");
  iconbtn.classList.toggle("icon-open");
}
```

```
// fonction d'opacité du menu au scroll
let prevScrollpos = window.scrollY;
window.onscroll = function() {
  let currentScrollPos = window.scrollY;
  if (prevScrollpos > currentScrollPos) {
    document.querySelector(".yellowMenu").style.opacity =
"1";
    document.querySelector(".blueBottom").style.opacity =
"1";
  } else {
    document.querySelector(".yellowMenu").style.opacity =
"0.1";
    document.querySelector(".blueBottom").style.opacity =
"0";
  }
  prevScrollpos = currentScrollPos;
};
```

Extrait Javascript 2: Afin d'avoir une meilleure lisibilité du site, il a fallu penser une fonction pour masquer le menu et le footer au scroll. Pour cela je déclare une première variable de position sur l'axe Y de la fenêtre du navigateur. Ensuite j'exécute une fonction onscroll sur cette fenêtre dans laquelle je re-déclare une variable de position sur l'axe des Y pour écrire

un « if statement » . Si ma souris scroll vers le bas, le menu et footer disparaissent sinon si je remonte ils réapparaissent.

3. Configuration YAML de l'interface EasyAdmin

Avec la découverte du bundle EasyAdmin, j'ai pu créer et détailler l'affichage du backOffice de mon application par le biais du fichier « easy_admin.yaml » situé dans le dossier de configuration. Ce fichier permet à EasyAdmin de savoir ce qu'il affiche suivant le niveau de permission des utilisateurs. C'est ici que l'on détaille et présente nos Entity avec les options qu'il est possible d'effectuer dessus. J'ai mis en place un menu de sélection de mes tables dans lequel je fais appel à une bibliothèque d'icône intégrée au bundle. J'ai édité mes entités afin de paramétrer la gestion du Crud ainsi que la lecture de certaines d'entre elles sous condition de permission. En YAML le respect de l'indentation est une obligation sinon ça ne fonctionne simplement pas.

Extrait Yaml 1: Dans cet extrait de code ci dessous je vous présente la réalisation du menu de sélection des entités du backOffice. Le label permet de donner une identification de celle-ci, dans mon cas cela me donne la possibilité de faire une traduction française du nom de mes tables. La permission d'accès au menu est accordée en fonction du rôle de l'administrateur. Un icône en rapport au label est choisi dans la bibliothèque « fontawesome » comprise dans le bundle. L'entity à relier est déclarée. Le menu est terminé.

```
menu:
  - { label: 'Administrateurs', permission: ROLE_SUPER_ADMIN,
    icon: 'user-circle-o', entity: 'Admin' }
  - { label: 'Plate-Forme', permission: ROLE_SUPER_ADMIN, icon:
    'fas fa-american-sign-language-interpreting', entity: 'Network' }
  - { label: 'Evènements', permission: ROLE_ADMIN, entity:
    'Event', icon: 'far fa-calendar-alt' }
  - { label: 'Associations', permission: ROLE_ADMIN, icon: 'fas
    fa-project-diagram', entity: 'Structure' }
  - { label: 'Référents', permission: ROLE_USER, icon: 'user',
    entity: 'Referent' }
  - { label: 'Discriminations type', permission: ROLE_ADMIN,
    icon: 'fas fa-book', entity: 'Discrimination' }
  - { label: 'Déclarations', permission: ROLE_SUPER_ADMIN,
    icon: 'fas fa-exclamation-circle', entity: 'Declaration' }
  - { label: 'Mots Clés', permission: ROLE_ADMIN, icon: 'fab
    fa-key', entity: 'Keyword' }
  - { label: 'Domaine', permission: ROLE_ADMIN, icon: 'fas fa-
    arrows-alt', entity: 'Domain' }
```

Extrait Yaml 2:

```
Admin:
  class: App\Entity\Admin
  controller: App\Controller\AdminController
  role: ROLE_SUPER_ADMIN

  list:
    title: 'Utilisateurs'
    actions:
      - { name: 'new', label: 'Ajouter', icon: 'add' }
      - { name: 'show', label: '', icon: 'search' }
      - { name: 'edit', label: '', icon: 'edit' }
      - { name: 'delete', label: '', icon: 'trash' }
    fields: &listUser
      - { property: 'id', label: 'ID' }
      - { property: 'lastname', label: 'Nom' }
      - { property: 'firstname', label: 'Prénom' }
      - { property: 'username', label: 'Login' }
      - { property: 'roles', label: 'Rôles' }
      - { property: 'enabled', label: 'Activé ?', type:
'toggle' }
    item_permission: ROLE_SUPER_ADMIN
  show:
    title: 'Fiche de l''utilisateur'
    fields: *listUser
    item_permission: ROLE_SUPER_ADMIN
  new:
    title: 'Ajouter un utilisateur'
    item_permission: ROLE_SUPER_ADMIN
  edit:
    title: 'Editer un utilisateur'
    item_permission: ROLE_SUPER_ADMIN
  form:
    fields:
      - { type: 'group', label: 'Infos', icon: 'info-
circle' }
      - { property: 'lastname', css_class: 'col-sm-6',
label: 'Nom' }
      - { property: 'firstname', css_class: 'col-sm-6',
label: 'Prénom' }
      - { type: 'group', css_class: 'col-sm-6', label:
'Identifiants', icon: 'key' }
      - { property: 'email', label: 'Login' }
      - { property: 'password', type: 'repeated',
type_options: { type:
'Symfony\Component\Form\Extension\Core\Type>PasswordType',
first_options: {label: 'Mot de passe' }, second_options:
{ label: 'Confirmer le mot de passe' }, invalid_message: 'Les
deux mots de passe ne concordent pas.' }}
      - { type: 'group', label: 'Permissions', icon:
'shield' }
      - {property: "roles", label: "Users roles", type:
"choice", type_options:{ expanded: true, multiple: true, choices:
{ "ROLE_SUPER_ADMIN": "ROLE_SUPER_ADMIN", "ROLE_ADMIN":
"ROLE_ADMIN", "ROLE_USER": "ROLE_USER" },},}
      - { type: 'group', label: 'Etat', icon: 'unlock' }
      - { property: 'enabled', label: 'Activé ?' }
    item_permission: ROLE_SUPER_ADMIN
```

Dans cet extrait nous avons le code de l'entité admin. Je donne le chemin de la classe entité Admin de mon application. J'indique le chemin du contrôleur qui implémente le fonctionnement du bundle et gère la création de nouveaux administrateurs ou utilisateurs. Je décris l'affichage en liste de ma table en lui donnant un titre, les champs qu'elle contient ainsi que la liste des actions représentées sous forme d'icône permettant d'effectuer le CRUD. Le langage yaml permet la mutualisation grâce aux caractères « & » et « * » ce qui me permet d'éviter la duplication de contenu. Les actions « list, show, edit, new », représentent différentes vues de ma table au sein d'EasyAdmin. Je paramètre la vue de la « list » en indiquant le contenu et le type des champs (fields) de ma table. Le type toggle permet l'apparition d'un bouton de type activation/désactivation pour tout champ à contenu booléen. L'option « item_permission » permet d'indiquer le niveau du rôle requis pour la lecture des items. Je mutualise le contenu de ma vue « list » dans ma vue « show ». Pour finir je mets en place le formulaire qui me permettra d'agir sur mes vues « new et edit » ce qui me permet d'ajouter une option dans la propriété « password » afin de faire appel à un composant de sécurité gérant le cryptage de celui-ci.

. 4. Display de compte utilisateur(admin) par implémentation de composant Symfony.

Pour pouvoir rendre le bouton d'activation de compte utilisable il a fallu implémenter le composant de sécurité « UserCheckerInterface », éditer un nouveau message d'exception relatif à la désactivation d'un compte en faisant une extension du composant de sécurité « AccountStatusException » et inscrire le nouveau composant dans le fichier security.yaml de l'application.

Extrait 1: extension du composant de sécurité « AccountStatusException ».

```
<?php
namespace App\Security;
use Symfony\Component\Security\Core\Exception\AccountStatusException;
class AccountDisabledException extends AccountStatusException
{
    /**
     * {@inheritdoc}
     */
    public function getMessageKey()
    {
        return 'Votre compte est désactivé.';
    }
}
```

Extrait 2: implémentation du composant de sécurité « UserCheckerInterface » pour insérer un « if statement » dans la fonction de vérification de pré-autorisation.

```
<?php
namespace App\Security;
use App\Entity\Admin as AppUser;
use Symfony\Component\Security\Core\User\UserCheckerInterface;
use Symfony\Component\Security\Core\User\UserInterface;

class UserChecker implements UserCheckerInterface
{
    public function checkPreAuth(UserInterface $user)
    {
        if (!$user instanceof AppUser) return;

        // L'utilisateur n'est pas activé par l'administrateur
        if (!$user->getEnabled()) {
            throw new AccountDisabledException();
        }
    }

    public function checkPostAuth(UserInterface $user)
    {
    }
}
```

Extrait 3: inscription du nouveau composant de sécurité « UserChecker » dans le fichier de configuration de l'application « security.yaml ».

```
firewalls:
  dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
  main:
    anonymous: true
    user-checker : App\Security\UserChecker
    guard:
      authenticators:
        - App\Security\AppAuthenticator
    logout:
      path: app_logout
```

5. Display d'affichage d'événement via Twig.

Les deux projets sur lesquels j'ai travaillé, nécessitaient une fonction d'activation pour afficher ou non un événement. Nous avons déjà vu qu'il était possible de générer un bouton d'activation dans la vue de table sur le backOffice à partir du moment où celle-ci dispose d'un champ de type booléen. Ici notre requête d'activation de ces données

événementielles ne concerne que leur affichage coté utilisateur anonyme. Pour effectuer cette opération, j'ai ajouté un « if statement » dans la template Twig de mes événements.

Extrait Twig: ajout du « if statement » dans ma template Twig.

```
<div class="row">
    {% for event in events %}
        {% if event.enabled == 1 %}
            <div class="col-12 col-md-6 col-lg-4 col-xl-4 d-flex
justify-content-around">
                <div class="eventCard p-3 mt-5">
                    <fieldset class="border p-3">
                        <legend class="w-auto">{{ event.title }}
</legend>
                        <div>
                            <div class="eventCard__img">
                                
                            </div>
```

Présentation d'un jeu d'essai

Un jeu d'essai est une description d'une suite d'actions et de résultats attendus, permettant la constitution de recettes de tests optimales. Les scénarios et les jeux d'essai sont réalisés avant le codage ou les tests d'une application. C'est un jeu de données couvrant le maximum de cas d'utilisations possibles permettant de tester une fonctionnalité. L'intérêt est de tester le maximum de cas de figures et en particulier les cas limites ou edge cases. Avant la mise en place des projets, les fonctionnalités critiques ont été relevées. L'une de ces fonctionnalités était la fonction de display de compte utilisateurs. Cette fonction doit permettre d'activer ou non un compte utilisateur afin d'en contrôler l'accès lors du login vers le backOffice. Je décide donc de réaliser un jeu d'essai sur cette fonctionnalité de display de compte complété au login.

Jeu d'essai

Fonctionnalité: Display compte utilisateur/Admin.

Cas d'utilisation (code + nom): C1 - Login d'accès au backOffice de l'application

Script SQL de la table Admin avec jeu de données:

```
CREATE TABLE `admin` (  
  `id` int(11) NOT NULL,  
  `email` varchar(180) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `roles` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `firstname` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `lastname` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `enabled` tinyint(1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;  
  
INSERT INTO `admin` (`id`,`email`,`roles`,`password`,`firstname`,`  
  `lastname`,`enabled`) VALUES  
(1, 'admin@admin.fr', '["ROLE_SUPER_ADMIN","ROLE_ADMIN",  
  "ROLE_USER"]',  
  '$argon2id$v=19$m=65536,t=4,p=1$T2hnRU91RUhSbEJFSkdiaQ$I8JtgEtgo7i7sMJtNI  
  BBgmHPJ/0NHQHHfHi0lbDe274', 'Root', 'Admin', 1),  
(2, 'toto@iswinner.fr', '["ROLE_ADMIN","ROLE_USER"]',  
  '$argon2id$v=19$m=65536,t=4,p=1$MTkveUtNN2wyYkdWVG9hVw$4zH9UmJ9/  
  H1LlqIInawGwWow+XFJbhEV74wn2S9VYYA', 'Winner', 'Toto', 0),  
(3, 'root@root.fr', '["ROLE_SUPER_ADMIN","ROLE_ADMIN"]', 'admin',  
  'Toto', 'NewAdmin', 1),  
(4, 'user@simple.fr', '["ROLE_USER"]',  
  '$argon2id$v=19$m=65536,t=4,p=1$Zy5xekxPRTJsUzZ2RHJqTQ$1qUoaNfE0A5zNZNZu7  
  hwm2c0uuUowBzAyzXhd3pbk+0', 'Use', 'simpleUser', 1),  
(5, 'user2@simple.fr', '["ROLE_USER"]',  
  '$argon2id$v=19$m=65536,t=4,p=1$ZXp3a3VzaGFsUVZLNHkDSg$ymJH/  
  x0yQRO+NDjUS0bnKJQRDXnJMUZKq5x+9wVivXs', 'Use2', 'simpleUser2', 0);
```

La connexion à l'espace d'administration doit se faire via un login mail ainsi qu'un mot de passe encrypté. Pour que la connexion soit optimale, le compte de l'utilisateur Administrateur doit être activé et le rôle minimum de connexion doit être admin.

ordre	Scénario : code + nom	Cas de test : valeurs	Cas de test : résultats attendus
1	C1-SN-Scénario nominal	utilisateur: Root Admin, rôle: ROLE_SUPER_ADMIN, ROLE_ADMIN, ROLE_USER, login de connexion: <u>admin@admin.fr</u> Password: encrypted password OK, isEnabled: 1	Accès à l'interface administrateur backOffice autorisé
2	C1-SE1-Scénario d'exception: compte désactivé.	utilisateur: Winner Toto, rôle: ROLE_ADMIN, ROLE_USER, login de connexion: toto@iswinner.fr Password: encrypted password OK, isEnabled: 0	Message: « votre compte est désactivé » Accès à l'interface administrateur bloqué
3	C1-SE2-Scénario d'exception: password invalide.	utilisateur: Toto NewAdmin, rôle: ROLE_SUPER_ADMIN, ROLE_ADMIN, login de connexion: toto@iswinner.fr Password: admin, isEnabled: 1	Message: « vous n'avez pas les autorisations requises » Accès à l'interface administrateur bloqué.
4	C1-SE3-Scénario d'exception: Rôle requis insuffisant.	utilisateur: Use simpleUser, rôle: ROLE_USER, login de connexion: toto@iswinner.fr Password: encrypted password OK, isEnabled: 1	Message: « vous n'avez pas les droits d'accès nécessaires » Accès à l'interface administrateur bloqué.
5	C1-SE4-Scénario d'exception: Rôle requis insuffisant.	utilisateur: Use2 SimpleUser2, rôle: ROLE_USER, login de connexion: toto@iswinner.fr Password: encrypted password OK, isEnabled: 0	Message: « vous n'avez pas les droits d'accès nécessaires » Accès à l'interface administrateur bloqué.

Autres cas possibles

ordre	Scénario : code + nom	Cas de test : valeurs	Cas de test : résultats attendus
6	C1-SE5-Scénario d'exception: compte non enregistré.	utilisateur: Unknow, rôle: no login de connexion: unknow@iswinner.fr Password: toto, isEnabled:no	Message: « votre email ne peut-être trouvé » Accès à l'interface administrateur bloqué
7	C1-SE6-Scénario d'exception: login invalide.	utilisateur: Unknow, rôle: no login de connexion: unknow Password: toto, isEnabled:no	Message: «Veuillez inclure un « @ » dans votre login » Accès à l'interface administrateur bloqué.
8	C1-SE7-Scénario d'exception: Mot de passe absent.	utilisateur: Unknow, rôle: no login de connexion: unknow@iswinner.fr Password: no isEnabled:no	Message: «Veuillez renseigner ce champs » Accès à l'interface administrateur bloqué.

Les injections SQL sont un ensemble de méthodes d'exploitation de faille de sécurité, d'une application fonctionnant avec un système de base de données. Ces injections profitent d'un manque de sécurisation des différents champs de connexion ou de formulaire. Il est possible, d'attaquer la requête SQL allant à la base de données dans le but d'injecter de nouvelles données étrangères ou non légitimes, ou passer de nouvelles commandes SQL. Le langage SQL étant un langage serveur celui-ci est compris par le serveur web. Les champs non protégés constituent dès lors une porte d'entrée pour les modifications de tout genre ainsi que le vol de données. Comme nous avons pu le voir dans le choix de notre stack technique, l'usage d'un framework permet une sécurisation optimisée. Nous allons donc observer comment et avec quelle mise en place lutter efficacement contre les injections SQL dans l'environnement de travail de Symfony.

1. Les attaques par injections SQL (SQLi) et Symfony.

Symfony intègre d'origine des fonctionnalités de sécurité comme la plupart des framework. La réputation robuste de Symfony ne veut pas dire que tout est déjà prévu et mis en place pour assurer l'optimisation de la sécurité de nos applications. Nos applications devant pouvoir transmettre des informations personnelles à caractère sensible il est important que celles-ci soient sécurisés. Une méthode assez simple dans Symfony est de mettre en place, des contraintes de validation sur les champs de nos formulaires. Un certain nombre de contraintes sont disponibles nativement dans Symfony. Il en existe environ une soixantaine répertoriée en neuf types différents. Ces contraintes vont avoir pour rôle de vérifier et d'encadrer des groupes de données ce qui empêchera l'utilisateur de fournir des données erronées ou d'insérer des caractères non autorisés dans les champs de formulaire. Par exemple, une contrainte de validation « Email » obligera l'utilisateur à entrer des données de type email contenant un "@" ainsi qu'un « . » dans la deuxième partie hôte de l'adresse mail. Une autre contrainte de validation « Regex » permettra elle d'empêcher, l'injection de tout type de caractère préalablement défini. Une contrainte de « date » pourra vérifier que celle-ci est valide etc etc... L'un des gros avantages est qu'un large choix de contraintes est déjà présent dans Symfony et il est

possible de pousser encore plus loin en éditant des contraintes personnalisées type custom.

Symfony est donc bien prévu avec des fonctionnalités de sécurités natives mais il est impératif pour compléter celles-ci de mettre en place des contraintes de validation. Dans le cas où ces mesures ne suffiraient pas à assurer l'intégrité de nos bases de données une solution supplémentaire doit être envisagée. En effet nos données, en plus d'être volées peuvent aussi être retenues en otage contre rançon ou voir même détruites. Pour répondre à ces situations une sauvegarde de la base de donnée peut être faite.

2. La sauvegarde de la base de données MySQL.

Plusieurs possibilités existent pour sauvegarder et restaurer une base de données. Depuis l'interface de PhpMyAdmin, nous pouvons importer ou exporter directement une base de données, une table ou un champ au format SQL, Csv, Json etc.. Le serveur MySQL fournit un programme client de sauvegarde de base de données, « mysqldump » s'exécutant directement en ligne de commande. Ce programme réalise une sauvegarde et génère une sortie au format SQL, Csv, autre texte délimité ou format XML. Afin de permettre un meilleur maintien que celui d'une sauvegarde manuelle, il est possible d'automatiser cette action à l'aide de « cron ». Cron est un programme disponible sur les systèmes de type UNIX (Linux, MacOS, FreeBSD). C'est un programme de planification de gestion et d'exécution automatique, de script, de commandes ou de logiciels à une date et une heure précise ou selon un cycle défini. Il est possible de combiner mysqldump avec cron pour automatiser nos sauvegardes. Avec une tâche cron (table de planification) cron va exécuter mysqldump qui réalisera la sauvegarde automatiquement. Beaucoup d'hébergements Web disposent de cron dans leur fonctionnement directement accessible depuis l'espace client. Par exemple sur OVH, une fonction de sauvegarde journalière de la base de données s'opère automatiquement. Il est possible si l'on souhaite augmenter la récurrence des sauvegardes dans la limite d'une par heure d'éditer une tâche cron en lui insérant un simple script PHP directement depuis l'interface de gestion client.

Comme on peut le constater, aujourd'hui, de plus en plus d'attention est portée sur la sécurisation de nos données. Beaucoup de nouvelles méthodes, contraintes et pratiques sont instaurées pour garantir la préservation de celle-ci. La sécurité est un apprentissage

en soi. Ces méthodes et contraintes ne suffisent pas sans mise en place préalable à assurer une sécurité optimale. L'importance de se mettre constamment à jour par le biais de veille numérique reste une nécessité pour la pérennité de nos développements.

Description d'une situation de recherche


Lors de la première semaine de développement du projet « Lutte contre les discriminations », nous nous sommes retrouvés confrontés à un premier problème. Je travaillais avec un autre apprenant sur la mise en place du bundle EasyAdmin et au moment du test de la création de nouveaux comptes admin, un résultat inattendu et non fonctionnel est survenu.


Les nouveaux comptes s'enregistrent bien en base de données mais l'étape d'encodage du mot de passe ne s'effectue pas. Les passwords sont inscrits en plainpassword ce qui rend les comptes inutilisables. En effet le composant de sécurité de Symfony attend lui une version cryptée des mots de passes enregistrés afin d'en effectuer sa vérification. Dans un premier temps, j'ai essayé de trouver la solution dans la recherche de bundle réalisant le travail. En parcourant la mine d'informations dont regorge le Web, je me suis rendu compte que plusieurs bundles d'administrations livrés avec des fonctionnalités de création d'utilisateur/Admin existaient. Un deuxième problème nous est tombé dessus celui cette fois de la version de Symfony. Aucune des solutions de type bundle que j'ai trouvé n'étaient compatible avec Symfony 5 (dernière version du framework). Dans cette nouvelle version, beaucoup de bundles auparavant utilisés ne remplissent plus les critères de sécurité liés au framework.

Cette nouvelle impasse, nous a obligé à intensifier les recherches de manières collectives en privilégiant la langue anglaise pour effectuer celle-ci. Aidé de notre formateur, un apprenant et moi-même nous nous sommes lancés cette recherche comme un challenge. Après plus d'une journée de recherche à écumer les sites internet, blogs et forums spécialisés comme StackOverflow, Supinfocom, etc. J'ai fini par trouver sur le github du bundle EasyAdmin, dans des échanges liés aux « issues » du bundle, une méthode pouvant résoudre notre problème. La réponse s'est donc trouvée dans l'override du bundle en lui créant une extension par le biais d'un nouveau contrôleur. Celui-ci permet d'ajouter, les nouvelles fonctions d'encodage password, avant qu'EasyAdmin ne l'enregistre en base de données ou lors de sa modification. La création de nouveaux administrateurs a enfin pu être possible.

Dans le domaine du développement Web, il n'est pas rare qu'un problème en cache un autre. Par chance, une énorme communauté active oeuvre sur tous les fronts et les développeurs sont rarement avarés en conseils. Plus les requêtes sont précises plus la solution devient proche. L'usage de l'anglais devient primordial pour augmenter les possibilités de résultats.


Traduction d'un extrait de site anglophone


 **Closed** password encryption #1261
kricha opened this issue on 4 Aug 2016 · 17 comments

 **javiereguiluz** commented on 4 Aug 2016 Collaborator ...


If I understand the code correctly, the problem is that EasyAdmin sees your Manager entity as any other entity. Therefore, it doesn't understand that a password field must be hashed before persisting it. For EasyAdmin this is just a normal form field.

If you use FOSUserBundle, [read this tutorial](#) about how to integrate it. Otherwise, you need to create your custom AdminController and override some method to call to the Symfony's password encoding service and save the change in the entity. This is way easier than it sounds! It's explained here in detail:
<https://github.com/javiereguiluz/EasyAdminBundle/blob/master/Resources/doc/book/7-complex-dynamic-backends.md>

 1

 **kricha** commented on 5 Aug 2016 Author ...

@javiereguiluz ok, but maybe it will be easier to create entity listener?

 **javiereguiluz** commented on 5 Aug 2016 Collaborator ...

@aLkRicha if you feel more comfortable with entity listeners, go for it! There's definitely more than one way to solve this.

However, "entity listeners" or "lifecycle callbacks" are considered a bad practice by Doctrine creators. See [these slides](#) for details.

The controller based solution feels a bit more "natural" for Symfony and it's a matter of adding a few lines of code.

In any case, let's close this issue as "fixed" because some valid solutions were proposed. Thanks!

Référence: <https://github.com/EasyAdminBundle/issues/1261>

Traduction:

_javiereguiluz a commenté le 5 aout 2016

Si je comprends le code correctement, le problème est que EasyAdmin voit votre entité administrateur comme n'importe quelle autre entité. Par conséquent, il ne comprend pas que c'est un champ de mot de passe qui doit être crypté avant d'être persisté en base

de données. Pour EasyAdmin c'est juste un champ normal. Si vous utilisez FosUserBundle, lisez ce tutoriel sur comment l'intégrer. Autrement vous devez créer votre contrôleur custom « AdminController » et implémenter quelques méthodes afin d'appeler le service d'encodage de mot de passe de Symfony et sauvegarder le changement dans l'entité. Cette procédure est plus facile qu'elle n'en a l'air. C'est expliqué ici en détails.

_kricha a commenté le 5 aout 2016

@javiereguiluz , d'accord mais peut-être, ce serait plus simple de créer un écouteur d'entité.

_Javiereguiluz a commenté le 5 aout 2016

@kricha si vous vous sentez plus confortable avec les écouteurs d'entités allez-y! Il y a définitivement plus qu'une manière pour résoudre cela. En revanche « les écouteurs d'entités et les rappels de cycle de vie » sont considérés comme une mauvaise pratique par les créateurs de Doctrine. Voir ces slides pour détail. La solution basée sur le contrôleur semble plus naturelle pour Symfony et ce n'est qu'un ajout de quelques lignes de code. Dans tous les cas, fermez le problème comme résolu car des solutions valides ont été proposées. Merci !

Conclusion

Cette période de notre formation, au départ incertaine m'a permis d'apprendre énormément, tant dans le développement Web que dans la relation humaine. Dans cette situation plutôt irréaliste, d'être en stage sans stage, j'ai réussi à l'aide, des autres apprenants, de notre formateur ainsi que de notre organisme de formation Utoplab à inverser le cours de notre histoire. Notre organisme a répondu à deux contrats grâce auxquels, j'ai décidé d'entreprendre, de fédérer et d'organiser notre équipe telle une agence Web. Ayant un niveau hétérogène, nous avons su nous répartir les tâches et nous donner les moyens d'avancer et d'apprendre ensemble, sans que personne ne soit lésé en terme d'expérience. Plusieurs points, comme l'apprentissage de Symfony par l'achat du dernier livre sur le développement du Framework ainsi que la mise en place d'une multitude d'autres outils ont été abordés. Chacune de nos idées pouvait trouver plusieurs solutions et il est devenu impératif de savoir analyser le besoin pour effectuer le meilleur choix.

Ce que je peux constater, est que l'on peut, facilement dire que rien n'est impossible dans le développement Web ce qui ne veut en aucun cas dire facile. Une connaissance de l'ordre du par coeur n'est pas utile. En revanche, faire preuve de méthodologie dans l'appréhension de chaque langage de programmation est pour moi une clé de réussite. Comprendre les principes de fonctionnement et les méthodes utilisés par la pratique, permet de passer plus aisément d'un langage à un autre sans crainte d'être perdu.

Le temps dans notre stage, nous a manqué ou alors le stage était trop court. L'énergie dégagée dans ce moment s'apparente aux principes de la communauté des développeurs Web. Ce stage m'a montré que l'on pouvait mourir devant son ordinateur, sans boire, ni manger et en ayant perdu la notion du temps. Défauts ou avantages, passion ou véritable drogue, il est important de faire preuve de pragmatisme dans mon apprentissage. Le développement Web est un mode de vie à part entière et l'immense communauté existante que l'on peut dire dévouée m'a permis et me permet encore de continuer à gravir les blocs de la programmation.

Bibliographie

Afin de réaliser ces projets et ce dossier j'ai consulté, analysé et appris d'un ensemble de recherche dont les sources sont listé ci-dessous:

Livres:

_En route vers Symfony 5

_Développer un site en PHP, MySQL, JavaScript, JQuery, Css3, Html5 (O'REILLY)

Liens de recherches Web:

[_https://devblog.lexik.fr/drupal/faciliter-la-gestion-des-droits-dacces-avec-les-voters-2488](https://devblog.lexik.fr/drupal/faciliter-la-gestion-des-droits-dacces-avec-les-voters-2488)

[_https://forum.alsacreations.com/topic-20-86499-1-Resolu-Symfony--User-Checkers.html](https://forum.alsacreations.com/topic-20-86499-1-Resolu-Symfony--User-Checkers.html)

[_https://stackoverflow.com/questions/52006043/symfony-easyadmin-choice-type-field-for-roles](https://stackoverflow.com/questions/52006043/symfony-easyadmin-choice-type-field-for-roles)

[_https://www.novaway.fr/notre-expertise/symfony](https://www.novaway.fr/notre-expertise/symfony)

[_https://www.sergentweb.com/guide-sur-les-prix-de-creation-d-un-site-internet/](https://www.sergentweb.com/guide-sur-les-prix-de-creation-d-un-site-internet/)

[_https://www.christophe-meneses.fr/article/bonnes-pratiques-symfony-organiser-sa-logique-metier](https://www.christophe-meneses.fr/article/bonnes-pratiques-symfony-organiser-sa-logique-metier)

[_https://thomas-boileau.fr/bonnes-pratiques-avec-symfony-4-et-la-poo/](https://thomas-boileau.fr/bonnes-pratiques-avec-symfony-4-et-la-poo/)

[_https://hackernoon.com/8-popular-php-frameworks-for-web-development-in-2020-od3f38ez](https://hackernoon.com/8-popular-php-frameworks-for-web-development-in-2020-od3f38ez)

[_https://symfony.com/legacy/doc/jobeeet/1_2/fr/04?orm=Propel](https://symfony.com/legacy/doc/jobeeet/1_2/fr/04?orm=Propel)

[_https://openclassrooms.com/fr/courses/1733551-gerez-votre-serveur-linux-et-ses-services/5236051-installez-le-serveur-web-le-plus-utilise-au-monde-apache](https://openclassrooms.com/fr/courses/1733551-gerez-votre-serveur-linux-et-ses-services/5236051-installez-le-serveur-web-le-plus-utilise-au-monde-apache)

[_https://fr.wikipedia.org/wiki/Apache_HTTP_Server](https://fr.wikipedia.org/wiki/Apache_HTTP_Server)

[_https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/](https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/)

[_https://fr.wikipedia.org/wiki/MySQL#Interop%C3%A9rabilit%C3%A9](https://fr.wikipedia.org/wiki/MySQL#Interop%C3%A9rabilit%C3%A9)

[_https://kinsta.com/fr/blog/comparaison-php/](https://kinsta.com/fr/blog/comparaison-php/)

[_https://w3techs.com/technologies/details/pl-php](https://w3techs.com/technologies/details/pl-php)

[_https://fr.wikipedia.org/wiki/PHP](https://fr.wikipedia.org/wiki/PHP)

[_https://openclassrooms.com/fr/courses/1885491-prenez-en-main-bootstrap/1885777-mise-en-route](https://openclassrooms.com/fr/courses/1885491-prenez-en-main-bootstrap/1885777-mise-en-route)

[_https://getcomposer.org/doc/00-intro.md](https://getcomposer.org/doc/00-intro.md)
[_https://connect.adfab.fr/outils/quest-ce-composer](https://connect.adfab.fr/outils/quest-ce-composer)
[_https://www.supinfo.com/articles/single/141-tout-savoir-bundle-symfony](https://www.supinfo.com/articles/single/141-tout-savoir-bundle-symfony)
[_https://www.grafikart.fr/tutoriels/bundles-280](https://www.grafikart.fr/tutoriels/bundles-280)
[_https://www.prestaconcept.net/blog/symfony/vichuploaderbundle-dry-configuration](https://www.prestaconcept.net/blog/symfony/vichuploaderbundle-dry-configuration)
[_https://www.lafabriquedunet.fr/blog/definition-api/](https://www.lafabriquedunet.fr/blog/definition-api/)
[_https://github.com/dustin10/VichUploaderBundle](https://github.com/dustin10/VichUploaderBundle)
[_https://symfony.com/doc/current/frontend.html](https://symfony.com/doc/current/frontend.html)
[_https://webpack.js.org/](https://webpack.js.org/)
[_https://afsy.fr/avent/2017/01-easyadminbundle-l-arriere-guichet-easy-peasy](https://afsy.fr/avent/2017/01-easyadminbundle-l-arriere-guichet-easy-peasy)
[_https://symfony.com/doc/current/frontend/encore/installation.html](https://symfony.com/doc/current/frontend/encore/installation.html)
[_https://github.com/symfony/webpack-encore](https://github.com/symfony/webpack-encore)
[_https://www.openaccess.fr/la-veille-du-web/easyadminbundle-une-interface-dadministration-symfony-decouvrir](https://www.openaccess.fr/la-veille-du-web/easyadminbundle-une-interface-dadministration-symfony-decouvrir)
[_https://symfony.com/doc/current/bundles/EasyAdminBundle/integration/vichuploaderbundle.html#uploading-other-types-of-files](https://symfony.com/doc/current/bundles/EasyAdminBundle/integration/vichuploaderbundle.html#uploading-other-types-of-files)
[_https://engineering.fb.com/web/yarn-a-new-package-manager-for-javascript/](https://engineering.fb.com/web/yarn-a-new-package-manager-for-javascript/)
[_https://www.blogduwebdesign.com/yarn-gestionnaire-de-paquets-pour-nodejs/](https://www.blogduwebdesign.com/yarn-gestionnaire-de-paquets-pour-nodejs/)
[_https://jqueryui.com/](https://jqueryui.com/)
[_https://github.com/tabalinas/jssocials](https://github.com/tabalinas/jssocials)
[_https://openclassrooms.com/fr/courses/1631636-simplifiez-vos-developpements-javascript-avec-jquery/1636887-jquery-ui](https://openclassrooms.com/fr/courses/1631636-simplifiez-vos-developpements-javascript-avec-jquery/1636887-jquery-ui)
[_https://github.com/jackducasse/caleandar](https://github.com/jackducasse/caleandar)
[_https://github.com/micku7zu/vanilla-tilt.js](https://github.com/micku7zu/vanilla-tilt.js)
[_https://fr.wikipedia.org/wiki/Document_Object_Model#:~:text=Le%20Document%20Object%20Model%20\(DOM,le%20contenu%20du%20navigateur%20web.](https://fr.wikipedia.org/wiki/Document_Object_Model#:~:text=Le%20Document%20Object%20Model%20(DOM,le%20contenu%20du%20navigateur%20web.)
[_https://micku7zu.github.io/vanilla-tilt.js/](https://micku7zu.github.io/vanilla-tilt.js/)
[_https://fr.wikipedia.org/wiki/Framework_JavaScript](https://fr.wikipedia.org/wiki/Framework_JavaScript)
[_https://www.ovh.com/fr/hebergement-web/technologie_crontab.xml](https://www.ovh.com/fr/hebergement-web/technologie_crontab.xml)
[_https://www.vaadata.com/blog/fr/la-securite-sur-symfony-episode-1-injection-xss-auth/](https://www.vaadata.com/blog/fr/la-securite-sur-symfony-episode-1-injection-xss-auth/)
[_https://fr.wikipedia.org/wiki/Injection_SQL](https://fr.wikipedia.org/wiki/Injection_SQL)
[_https://symfony.com/doc/current/reference/constraints.html](https://symfony.com/doc/current/reference/constraints.html)
[_https://symfony.com/doc/current/reference/constraints/Email.html](https://symfony.com/doc/current/reference/constraints/Email.html)
[_https://symfony.com/doc/current/reference/constraints/NotBlank.html](https://symfony.com/doc/current/reference/constraints/NotBlank.html)

[_https://doc.ubuntu-fr.org/cron](https://doc.ubuntu-fr.org/cron)
[_https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/mysqldump-sql-format.html](https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/mysqldump-sql-format.html)
[_https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/backup-methods.html](https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/backup-methods.html)
[_https://community.jaguar-network.com/sauvegarde-automatique-dune-db-mysql-avec-mysqldump/](https://community.jaguar-network.com/sauvegarde-automatique-dune-db-mysql-avec-mysqldump/)
[_https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html)
[_https://www.kinamo.fr/fr/support/faq/mysql-backup-automatique-base-de-donnees](https://www.kinamo.fr/fr/support/faq/mysql-backup-automatique-base-de-donnees)
[_https://fr.wikipedia.org/wiki/Cron](https://fr.wikipedia.org/wiki/Cron)
[_https://docs.ovh.com/fr/hosting/mutualise-taches-automatisees-cron/](https://docs.ovh.com/fr/hosting/mutualise-taches-automatisees-cron/)
[_http://www-igm.univ-mlv.fr/~dr/XPOSE2000/TesTs/SiteWeb/constests.htm#_Plan_d%E2%80%99un_jeu](http://www-igm.univ-mlv.fr/~dr/XPOSE2000/TesTs/SiteWeb/constests.htm#_Plan_d%E2%80%99un_jeu)
[_https://mlopes2016.files.wordpress.com/2017/12/jeu_d_essai.pdf](https://mlopes2016.files.wordpress.com/2017/12/jeu_d_essai.pdf)
[_https://symfony.com/doc/current/security/user_checkers.html](https://symfony.com/doc/current/security/user_checkers.html)
[_http://www-ftp.lip6.fr/ftp/lip6/reports/1997/lip6.1997.038-18.pdf](http://www-ftp.lip6.fr/ftp/lip6/reports/1997/lip6.1997.038-18.pdf)
[_https://www.akuiteo.com/blog/jeu-dessai-etape-choix-dun-erp-demo-logiciel-de-gestion](https://www.akuiteo.com/blog/jeu-dessai-etape-choix-dun-erp-demo-logiciel-de-gestion)
[_https://phortail.org/webntic/Comment-tester-une-application-avec-un-jeu-d-essai.html](https://phortail.org/webntic/Comment-tester-une-application-avec-un-jeu-d-essai.html)
[_https://meillierportefeuelles.files.wordpress.com/2015/02/plan-de-test.pdf](https://meillierportefeuelles.files.wordpress.com/2015/02/plan-de-test.pdf)
[_https://books.google.co.uk/books?id=U5ckDwAAQBAJ&pg=PA176&lpg=PA176&dq=Jeu+d%27essais+et+v%C3%A9rifications&source=bl&ots=OmeLI5o62K&sig=ACfU3U35cl6zjK5tuX-YxdCko3i9UT-WMg&hl=en&sa=X&ved=2ahUKEwjtn-2T6ufpAhX1ZxUIHU5TBkAQ6AEwBnoECAkQAQ#v=onepage&q=Jeu%20d'essais%20et%20v%C3%A9rifications&f=false](https://books.google.co.uk/books?id=U5ckDwAAQBAJ&pg=PA176&lpg=PA176&dq=Jeu+d%27essais+et+v%C3%A9rifications&source=bl&ots=OmeLI5o62K&sig=ACfU3U35cl6zjK5tuX-YxdCko3i9UT-WMg&hl=en&sa=X&ved=2ahUKEwjtn-2T6ufpAhX1ZxUIHU5TBkAQ6AEwBnoECAkQAQ#v=onepage&q=Jeu%20d'essais%20et%20v%C3%A9rifications&f=false)
[_https://www.applyss.com/2020/03/17/easyadmin-symfony-encode-users-passwords/](https://www.applyss.com/2020/03/17/easyadmin-symfony-encode-users-passwords/)
[_https://stackoverflow.com/questions/54744836/symfony-4-easyadmin-how-to-encrypt-passwords](https://stackoverflow.com/questions/54744836/symfony-4-easyadmin-how-to-encrypt-passwords)
[_https://github.com/EasyCorp/EasyAdminBundle/issues/1261](https://github.com/EasyCorp/EasyAdminBundle/issues/1261)
[_https://en.wikipedia.org/wiki/Edge_case](https://en.wikipedia.org/wiki/Edge_case)
[_https://fr.wikipedia.org/wiki/M%C3%A9thode_agile](https://fr.wikipedia.org/wiki/M%C3%A9thode_agile)