

Antoine Piron

Dossier Projet

# Reskyōō



Du 14 Novembre 2022 au 13 Janvier 2023

# **Sommaire**

<b>1. Introduction</b>	<b>2</b>
1.1 Remerciement	2
1.2 Compétences du référentiel couvertes par les projets	3
<b>2. Présentation Projets</b>	<b>4</b>
<b>3. Cahier des charges des projets</b>	<b>5</b>
3.1 Reskyoo	5
3.1.1 Besoins exprimés	5
3.1.2 Technologie choisies et employées	5
3.2 Popcode	6
3.2.1 Besoins exprimés	6
3.2.2 Technologies choisies et employées	6
<b>4. Réalisations significatives</b>	<b>7</b>
4.1 Envoi / création sur le serveur distant d'un objet possédant une ou plusieurs images ( Projet Reskyoo )	7
4.2 Recherche de point de retrait en fonction d'une adresse ou d'une position GPS demandé puis traitement de la liste pour correspondre aux critères demandés ( Projet Reskyoo )	9
4.3 Écouteur d'événement Javascript sur le clavier pour entrer les réponses utilisateurs ( Projet Popcode )	14
<b>5. Jeu d'essai de la fonctionnalité la plus représentative</b>	<b>19</b>
<b>6. Description de la veille de sécurité</b>	<b>22</b>
<b>7. Situation de travail ayant nécessité une recherche en anglais</b>	<b>24</b>
<b>8. Extrait d'un site utilisé lors des recherches en anglais accompagné d'une traduction.</b>	<b>24</b>
<b>9. Conclusion</b>	<b>26</b>

# 1. Introduction

## 1.1 Remerciement

Je me dois de remercier pas mal de personnes.

Tout d'abord, Loïc Dauchy , pour m'avoir formé pendant une précédente formation puis m'avoir accepté en stage. Le projet principal du stage ( Reskyoo ) a été un projet intéressant à développer et a permis de m'intéresser à plus de nouvelles technologies. Ma conversion totale à Symfony est probablement en marche grâce à ça.

Ensuite, Alain Merucci, notre formateur pour la formation "Développeur Web et Web Mobile", pour avoir réussi à maintenir ma motivation jusqu'au bout de la formation malgré certains événements. Ses projets tout au long de la formation agrémentés de certains challenges ont toujours réussi à sortir le meilleur de nous même pour apprendre et développer des choses.

Je n'oublie pas l'AccessCodeSchool et Online Formapro pour m'avoir donné cette chance de pouvoir quelque chose que j'apprécie.

Et pour finir , mes camarades de formation, toujours là pour passer de bons moments tout en se formant. Nos différences nous ont permis de penser à d'autres façons de développer ou d'autres façons de procéder dans notre raisonnement.

Je vous souhaite une bonne lecture de ce dossier.

## 1.2 Compétences du référentiel couvertes par les projets

Activités-types	Compétences professionnelles	
Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	Maquetter une application	
	Réaliser une interface utilisateur web statique et adaptable	✓
	Développer une interface utilisateur dynamique	✓
	Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce	
Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	Créer une base de données	✓
	Développer les composants d'accès aux données	✓
	Développer la partie back-end d'une application web ou web mobile	✓
	Elaborer et mettre en oeuvre des composants de gestion de contenu ou e-commerce	✓

## 2. Présentation Projets

Ce stage de 2 mois s'est déroulé en 2 temps pour moi, une période de 2 semaines sur un projet puis le reste du stage sur l'autre projet.

Dans un premier temps , un projet de rédaction de documentation à l'intention des développeurs pour la poursuite de développement et la maintenance d'une application. Cela m'a permis notamment de travailler avec le framework Ionic ainsi qu'apprendre à me débrouiller à la compréhension du code produit par une autre personne.

Dans un second temps , le projet Reskyoo nous a été attribué à Hamza MANAI (autre stagiaire de la formation) ainsi qu'à moi-même. Reskyoo est une application de récupération / achats à moindres coûts d'objets vendables mais non commercialisables dû à certaines imperfections. Pour le moment cette application n'est pas encore sortie, étant encore au stade de développement.

Le projet a porté sur la création d'une API pour cette application basée sur Symfony et API platform afin de pouvoir facilement relier l'application aux différents ERP (Enterprise Ressource Planning) des entreprises concernant la gestion de leurs stocks. J'ai couvert la partie création de facturation , géolocalisation / calcul de distance entre deux points donnés et enfin le début de la partie paiement avec Stripe. Outre Symfony et API Platform, j'ai été amené à utiliser Doctrine comme ORM ainsi que Geoapify pour la récupération de coordonnées GPS.

Je ne présenterai pas le premier projet ici , car il ne couvre que très peu du référentiel et relève surtout des bonnes pratiques que doit avoir un développeur.

Pour la partie Front end , je présenterai donc un projet réalisé dans le cadre de la formation en centre nommé Popcode ( accessible à l'adresse [popcode.a-piron.fr](http://popcode.a-piron.fr) ) . Il s'agit d'un jeu en navigateur web de recherche , basé sur Popcorn Garage et Popcorn TV. Le but du jeu est de retrouver diverses références à des langages de programmations cachés dans une image et les rentrer sur son clavier. Le projet a été mis en place via du Javascript ainsi que du HTML et du CSS. Certains petit défis ont été ajouté pour augmenter la difficulté du développement comme par exemple une acceptation de la réponse si celle ci est à 80% bonne ( Nombres de lettres , ordres , etc ) ou même un petit système de sauvegarde.

## 3. Cahier des charges des projets

### 3.1 Reskyoo

#### 3.1.1 Besoins exprimés

Il avait été exprimé le besoin de produire une API ( Application Programming Interface ) pour l'application afin de permettre une coopération facilitée avec les différents ERP ( Enterprise Resource Planning ou Progiciel de gestion intégré ). L'API devait pouvoir:

- Créer des utilisateur ainsi que leur permettre de se connecter,
- Permettre aux utilisateurs de faire une requête de mot de passe perdu,
- Pouvoir enregistrer des produits avec une ou plusieurs images ,
- Pouvoir enregistrer de nouveaux points de retraits avec des images,
- Effectuer une recherche sur les points de retraits et / ou produits selon divers critères,
- Créer des transactions en base de données ainsi que des factures associées afin de gérer un meilleur suivis par entreprise,
- Avoir des statistiques par utilisateurs couvrant les économies que celui-ci à réaliser,
- Avoir des statistiques pour les entreprises couvrant les revenus par années, les revenus par mois, les produits les plus vendus , les catégorie de produits les plus vendus , les clients achetant le plus souvent et le nombre de transactions par mois,
- Plus d'informations concernant les entreprises ( le siret, le nom et la descriptions)

#### 3.1.2 Technologie choisies et employées

Pour ce projet , les technologies choisies sont :

- Symfony , framework imposé par le maître de stage pour des raisons de facilités de développement en raison des différents bundles utilisé après,
- API Platform, bundle de gestion d'API pour Symfony, choix imposé par le maître de stage,
- Doctrine, l'ORM ( Object-Relational Mapping ou mapping objet-relationnel en français) de Symfony pour les relations avec la base de données
- Geoapify, API de géolocalisation et de calcul d'itinéraires, pour les besoins d'un service de récupération de coordonnées gps en fonction d'une adresse, j'ai choisi cette API car j'étais déjà familier avec son fonctionnement.
- FPDF, librairie de création de PDF , Choix fait après quelques recherches en la couplant à un complément facilitant son utilisation.

- PDF Invoice , complément de FPDF facilitant son utilisation , réaliser par artknonket sur Github
- Insomnia, logiciel de débbug / test d'API, pas de raisons particulières pour ce choix à par une recommandation de la part de notre maître de Stage

## 3.2 Popcode

### 3.2.1 Besoins exprimés

Le but de ce projet est de créer une application web ludique, permettant aux utilisateurs

par le biais d'un système de recherche et trouve, de découvrir différents langages informatiques de programmation et d'en savoir plus sur ceux-ci.

Le site devait être accessible sur ordinateur mais pour des raisons de gain de temps et de facilités, celui-ci ne devait pas être disponible sur mobile. En gérant les media queries, un message devait apparaitre afin d'indiquer à l'utilisateur que le jeux n'était disponible que sur ordinateur fixe

La maquette était déjà fournie en avance.

### 3.2.2 Technologies choisies et employées

Pour la partie statique et visuel , un mix entre du HTML et du CSS sera utilisé.

Pour la partie interactive, du Javascript natif sera retenu car une contrainte avait été ajoutée et était de n'utiliser aucun framework.

La librairie "Zoom by Ironex" sera utilisé afin de faciliter la gestion du zoom au sein du jeu

La maquette fournie se trouvait sur Figma.

## 4. Réalisations significatives

### 4.1 Envoi / création sur le serveur distant d'un objet possédant une ou plusieurs images ( Projet Reskyoo )

La première tâche qui m'a été demandé de réaliser est l'enregistrement d'une ou plusieurs images avec un produit lors de la création de celui-ci.

La première contrainte qui est apparue se trouvait être l'impossibilité d'utiliser les variables FILES car l'origine des demandes à l'API ne proviendrait pas forcément du même serveur ou celle-ci serait hébergée. La solution trouvée a été de transformer celle où ces dernières dans le format Base64 et d'envoyer le résultat obtenu sous forme de tableau rassemblant toutes les images transformées à l'API. Il s'agit du seul moment où j'ai dû travailler sur un petit fichier Javascript ainsi qu'un petit fichier HTML afin de pouvoir tester le fonctionnement de cette partie au lieu d'utiliser Insomnia.

Voici la fonction réalisée en javascript:

```
document.querySelector('#fileRegister').addEventListener('click', function(e){
    e.preventDefault();
    let xhr = new XMLHttpRequest();
    let reader = new FileReader();
    reader.readAsDataURL(document.getElementById('fileTest').files[0]);
    reader.onloadend = () => {
        let base64 = reader.result
        console.log(base64);
        xhr.open('POST', 'link to the api for new product');
        let testData = {
            name:"Test",
            description:"Velit sit iriure at et sanctus et lorem takimata",
            detail:"Invidunt dolor nostrud et nulla",
            category:"/api/categories/3",
            user:"/api/users/1",
            price:"29.99",
            defaultTypes:["/api/default_types/1"],
            reduction: "/api/reductions/1",
            wdPoint: "/api/wd_points/3",
            base64Image: [base64]
        };
        xhr.setRequestHeader('Authorization', 'Bearer token');
        xhr.setRequestHeader('Content-Type', 'application/json');
        xhr.send(JSON.stringify(testData));
    };
});
```



Concernant cette fonction, il s'agit d'une fonction classique combinant un objet de type `FileReader` pour la transformation d'un fichier en Base64, et un objet `XMLHttpRequest` afin d'envoyer les informations à l'API.

Le gros du travail de cette fonctionnalité réside dans la partie présente du côté de l'API.

Lors de l'appel à l'API, un fichier est lancé et une fonction démarre.

```
public function __invoke($data)
{
    $img = $data->getBase64Image();
    foreach ($img as $key) {
        $newImgAdd = $this->saveImage($key);
        $data->addImage($newImgAdd);
    }

    $this->em->persist($data);
    $this->em->flush();

    return [
        'product' => $data,
        'message' => 'success',
    ];
}
```

Cette fonction ne sert qu'à lancer d'autres fonctions et enregistrer l'image sur le serveur. La première fonction lancée sert à récupérer le tableau contenant les valeurs en Base64 des images. Celles-ci seront exploitées juste après dans une boucle sur le tableau. La première fonction employée dans la boucle est la suivante:

```
public function saveImage($base64Img){
    $img = explode( ' ', $base64Img );
    $name = md5(uniqid());
    $ext = '.'.explode('; ', explode('/', $img[0])[1] )[0];
    $destination = fopen($this->getUploadDir().$name.$ext, 'wb');
    fwrite($destination, base64_decode($img[1]));
    fclose($destination);

    $image = new Image();
    $image->setName($name.$ext);

    return $image;
}
```

Cette fonction prend un seul paramètre, une valeur en Base64. Cette valeur sera divisée plusieurs fois afin d'extraire l'information de l'extension du fichier envoyé. Un nouveau sera généré aléatoirement, l'encodage en md5 servant surtout à contrer le souci des 5 premières valeurs générées dans un `uniqid` étant toujours identique.

Un fichier sera ensuite ouvert avec le nom généré aléatoirement et l'extension récupéré dans le dossier d'upload en se servant du fait que si la fonction "fopen" ne trouve pas le fichier, celle-ci créera le fichier. Ensuite la seconde partie de la valeur de l'image sera inscrite dans le fichier.

Pour finir le fichier sera fermé afin de ne pas garder d'instance d'écriture ouverte et donc réduire la charge sur le serveur.

Une fois cela fait, on crée un nouvel objet image dans lequel on appelle la fonction pour enregistrer le nom. Pour finir, on retourne l'objet image à la fonction "\_\_invoke".

Pour finir, l'objet retourné sera utilisé dans la fonction "AddImage" qui permettra de faire le lien entre les tables de la base de données "image" et "product".

Les données envoyées à l'origine seront ensuite traitées afin de créer le produit en lui-même.

## 4.2 Recherche de point de retrait en fonction d'une adresse ou d'une position GPS demandé puis traitement de la liste pour correspondre aux critères demandés (Projet Reskyoo)

Cette partie est dispatché entre trois fichiers:

- un fichier repository pour les points de retraits,
- un contrôleur dédié à cette recherche se lançant et traitant la demande lors de l'appel de l'API
- un fichier service dédié à la partie géolocalisation / calcul de distances

Voici la fonction qui se lance à l'appel de l'API:

```
public function __invoke($data)
{
    $filterToCheck = [
        "levelReduc" => $data->getLevelReductionFilter(),
        "category" => $data->getCategoryFilter(),
        "priceMin" => $data->getPriceMinFilter(), // You, last month * update filter still need some fix
        "priceMax" => $data->getPriceMaxFilter(),
        "order" => $data->getOrderFilter()
    ];

    if ($data->getAddress() != "" && $data->getLat() == "" && $data->getLon() == "" && $data->getDep() == "") {
        $swdPointsToCheck = $this->geoService->getPointsByGeo($data->getLimitKlm(), [$data->getAddress()]);
        $swdPoints = $this->filtering($swdPointsToCheck, $filterToCheck);
        $response = 'success1';
    } else if ($data->getAddress() == "" && $data->getLat() != "" && $data->getLon() != "" && $data->getDep() != "") {
        $moreArg = array($data->getLat(), $data->getLon(), $data->getDep());
        $swdPointsToCheck = $this->geoService->getPointsByGeo($data->getLimitKlm(), $moreArg);
        $swdPoints = $this->filtering($swdPointsToCheck, $filterToCheck);
        $response = 'success2';
    } else {
        $swdPoints = '';
        $response = 'error missing arguments ( adress or lat/lon/dep) or isAddress = true but no address gave';
    }

    return [
        'product' => $swdPoints,
        'message' => $response,
    ];
}
```

Elle va d'abord initialiser un tableau qui va définir les différents paramètres de recherches en plus de la distance demandée par l'utilisateur. Puis, à l'aide d'une condition, vas vérifier si une adresse est envoyée ou si , au contraire, des coordonnées gps sont envoyées. Dans tous les cas , la fonction "getPointByGeo" du service Geo sera lancée puis les données récupérées de cette fonction seront utilisées dans une nouvelle fonction. La fonction "getPointByGeo" sera détaillée plus loin dans ce dossier.

La fonction de filtres nommé "filtering" prend deux paramètres:

- un tableau contenant tous les points de retraits
- un tableau contenant tous les paramètres de recherches

```
private function filtering($wdPointToCheck, $filterToCheck){
    $numberFilterToCheck = 0;
    $arrayProductToReturn = array();
    if ($filterToCheck['levelReduc'] != "") { ... You, last month * update filter still need some fix ... }
    if ($filterToCheck['category'] != "") { ...
    if ($filterToCheck['defaultType'] != "") { ...
    if ($filterToCheck['priceMin'] != "") { ...
    if ($filterToCheck['priceMax'] != "") { ...
    foreach ($wdPointToCheck as $key) {
        $products = $this->productRepo->GetByWdPoint($key['id']);
        foreach ($products as $product) {
            $priceAfterReduc = $product->getPrice() - (($product->getPrice()/100)*$product->getReduction()->getPercent());
            $checking = 0;
            if($filterToCheck['levelReduc'] == $product->getReduction()->getId() && $filterToCheck['levelReduc'] != ""){ ...
            if($filterToCheck['category'] == $product->getCategory()->getId() && $filterToCheck['category'] != ""){ ...
            if($filterToCheck['priceMin'] <= $priceAfterReduc && $filterToCheck['priceMin'] != ""){ ...
            if($filterToCheck['priceMax'] >= $priceAfterReduc && $filterToCheck['priceMax'] != ""){ ...
            if($filterToCheck['defaultType'] == $product->getDefaultTypes() && $filterToCheck['defaultType'] != ""){ ...
            if($checking == $numberFilterToCheck){
                $translate = array($product);
                array_push($arrayProductToReturn, $translate);
            }
        }
    }
    return $arrayProductToReturn;
}
```

Plusieurs conditions vont vérifier les divers paramètres de recherches et incrémenter une valeur de contrôle. Puis vont se lancer deux boucles, l'une dans l'autre. Pour chaque élément du tableau de points de retraits valide, l'id du dit point de retrait sera récupéré et utilisé avec une fonction de récupération de tous les produits.

La seconde boucle se déclenche alors et vérifie chaque produit récupéré si ceux-ci respectent les conditions de recherche. Si toutes les conditions sont remplies par rapport à celles demandées par l'utilisateur , le produit sera ajouté à un tableau et la vérification continuera jusqu'à la fin du contrôle de tous les produits d'un point de retrait.

Une fois la liste des points de retraits valide entièrement vérifiés et les produits valide ajoutés à un tableau , la fonction retournera ce tableau , qui sera ensuite retourné par la fonction "\_\_invoke".

Le service Geo rassemble diverses fonctions concernant la géolocalisation et le calcul de distance. C'est aussi dans ce service que sera fait l'appel à l'API Geoapify.

La fonction principale de ce service est une fonction de calcul de distance entre deux coordonnées gps.

```
public function getPointsByGeo($limitKlm, $moreArg){
    $allWdPoint = [];
    $latStart = "";
    $lonStart = "";
    $depStart = "";
    $toCheck = count($moreArg);
    switch ($toCheck) { ...
    }
    $deplist = $this->getGroupDep($depStart);
    foreach ($deplist as $dep) {
        $allWdPoint = array_merge($allWdPoint, $this->WdPointRepository->getByGroupDep($dep));
    }
    $validWdPoint = array();
    foreach ($allWdPoint as $row) {
        $latEnd = $row->getLat();
        $lonEnd = $row->getLon();
        $depEnd = $row->getDep();
        $result = $this->calcDist($latStart, $latEnd, $lonStart, $lonEnd);
        if ($result <= $limitKlm) {
            $newRow['address'] = $row->getAddress();
            $newRow['user'] = $row->getUser();
            $newRow['id'] = $row->getId();
            array_push($validWdPoint, $newRow);
        }
    }
    return $validWdPoint;
}
```

Il s'agit d'une fonction prenant en compte deux paramètres:

- une limite de kilométrage que l'utilisateur entrera,
- un tableau contenant les coordonnées gps ainsi que le département ou à défaut une adresse géographique.

Diverses variables sont créées et initialisées dans la fonction car elles ne serviront qu'ici.

Un switch est présent pour définir si le tableau d'argument passé ne contient qu'une adresse géographique ou si celui-ci contient les coordonnées GPS.

```
switch ($toCheck) {
    case 1:
        $requestLatLonDep = $this->getGeo($moreArg[0]);
        $latStart = $requestLatLonDep['latStart'];
        $lonStart = $requestLatLonDep['lonStart'];
        $depStart = $requestLatLonDep['depStart'];
        break;
    case 3:
        $latStart = $moreArg[0];
        $lonStart = $moreArg[1];
        $depStart = $moreArg[2];
        break;
    default:
        return 'Too Many Argument';
        break;
}
```

Selon le nombre d'entrées présentes dans le tableau , le switch effectuera ou non un appel à une autre fonction. La fonction appelée est aussi une fonction appellable à part , servant notamment lors de l'enregistrement d'un nouveau point de retrait.

```
public function getGeo($query){
    $dataReceive = json_decode(file_get_contents('https://api.geoapify.com/v1/
    geocode/search?text='.$query.'&format=json&
    apiKey=4ed6b9949b3a40268eeac014c0673286'), true);
    $lat = strval($dataReceive['results'][0]['lat']);
    $lon = strval($dataReceive['results'][0]['lon']);
    if (substr($dataReceive['results'][0]['postcode'], 0, -3) === "97") {
        $dep = substr($dataReceive['results'][0]['postcode'], 0, -2);
    }else{
        $dep = substr($dataReceive['results'][0]['postcode'], 0, -3);
    }

    $returnResult = [
        "latStart" => $lat,
        "lonStart" => $lon,
        "depStart" => $dep
    ];
    return $returnResult;
}
```

Le but de cette fonction est de faire appel à Geoapify afin de récupérer les coordonnées GPS d'une adresse particulière ainsi que son département. Une condition à été ajoutée lors de la récupération avant le renvoi de données afin de ne récupérer que le département et pas le code postal complet. De plus , cette condition permet aussi de différencier les numéros de département métropolitain ( 2 chiffres ) et les numéros de département d'outremer ( 3 chiffres ) dans les codes postaux.

Une fois la comparaison et le retour via le switch fait, on récupère les départements limitrophes en plus du département de l'adresse ou des coordonnées ciblé via la fonction "getGroupDep".

A partir de cette liste de département , on va cette fois appeler une fonction "getByGroupDep" dans une boucle qui va récupérer tous les points de retraits des département demandés et fusionner les retour dans un seul et unique tableau.

Une nouvelle variable sera initialisée et une nouvelle boucle sera lancée , et tournera tant qu'il y a des éléments dans le tableau qui n'auront pas été vérifiés. A chaque itération de la boucle , les coordonnées gps du point de départ et du point d'arrivée seront utilisées comme paramètres pour la fonction "calcDist".

```

private function calcDist($p1Lat, $p2Lat, $p1Lon, $p2Lon){
    $R = 6371; // earth's mean radius in km
    $dLat = $this->radians($p2Lat - $p1Lat);
    $dLong = $this->radians($p2Lon - $p1Lon);

    $a = sin($dLat/2) * sin($dLat/2) + cos($this->radians($p1Lat)) * cos($this->radians($p2Lat)) * sin($dLong/2) * sin($dLong/2);
    $c = 2 * atan2(sqrt($a), sqrt(1-$a));
    $d = $R * $c;

    return number_format($d,3);
}

```

Il s'agit de calcul de distance entre deux coordonnées gps. Cette fonction retourne un chiffre ayant jusqu'à 3 décimales et se trouvant être la distance en kilomètres entre les deux points.

En fonction du résultat, une condition vérifie si la distance entre le point de retrait et le point de départ et si la condition est valide, l'adresse, l'id de celui-ci et le compte utilisateur lié au point de retrait ( l'entreprise à qui ce point appartient ) sont ajouté à la liste des points de retraits valide pour la demande de l'utilisateur.

## 4.3 Écouteur d'événement Javascript sur le clavier pour entrer les réponses utilisateurs ( Projet Popcode )

Cet écouteur d'événement est l'élément central de Popcode car c'est de lui que découle quasiment toutes les interactions utilisateurs ainsi que les vérifications des réponses.

```
window.addEventListener('keydown', function checking(e) {
>   if (modalLegals.classList.contains('none') === false) { ...
>   }
>   if (mainGame.classList.contains('none')) { ...
>   }
>   if (modalDesc.classList.contains('none') === false) { ...
>   }
>   if (modallanguage.classList.contains('none') === false) { ...
>   }
    if (Input.textContent !== '') {
        if (e.code === "Enter"){
            clearTimeout();
            minInput = Input.textContent.toLowerCase();
            checkCountRotate = 0;
>           for (let i = 0; i < guessing.length; i++) { ...
>           }
>           if (checkCountRotate === guessing.length) { ...
>           }
        }
        if (e.code === 'Backspace'){
            Input.textContent = Input.textContent.slice(0, -1);
            return;
        }
    }
    for (let z = 0; z < letters.length; z++) {
        if (e.code === "Key"+letters[z] || e.code === letters[z] || e.key === '-' || e.key === '#' || e.key === '+') {
            if(overlay.classList[0] === 'none'){
                clearTimeout()
                removenone();
            }
            document.querySelector('#Input').textContent += e.key
            return;
        }
    }
})
```

L'écouteur s'ouvre sur 4 conditions vérifiant la présence ou non de certaines classes. Leur rôle n'est que pour du contrôle et de la correction d'interface. Les deux premières bloquent l'écouteur si l'utilisateur se trouve sur la page des mentions légales ou que celui-ci n'as pas démarré de partie. Les deux suivants vérifient si les modales de descriptions ou de langages trouvés sont ouvertes.

Vient ensuite une condition qui vérifient si le contenu de la réponse n'est pas vide et si c'est le cas , si certaines touches spécifiques n'ont pas été pressées. Cette condition et ces sous-conditions seront abordées plus loin dans le dossier.

Se lance ensuite une boucle faisant un nombre d'itération équivalent à la taille d'un tableau contenant toutes les lettres afin de permettre plus de flexibilités si l'on souhaite ajouter des chiffres ou même les variantes avec accents de certaines lettres. A chaque itération de la boucle une condition va vérifier la correspondance entre le code de la touche appuyé et sa présence dans l'élément du tableau actuellement vérifié. Il vérifie en même temps si certains caractères spéciaux ont été pressés. Si une correspondance est trouvée, la lettre sera ajoutée dans l'emplacement de la réponse et mettra fin à la boucle. Dans cette boucle se trouve aussi un retrait de tous les timeout étant actuellement en route, il s'agit là surtout d'une simple sécurité pour moi. Enfin si l'overlay permettant de visualiser l'écriture de la réponse pour l'utilisateur n'est pas affiché, celui sera affiché.

Si la zone de réponse contient du texte, la condition précédant la boucle d'écriture de lettres va se déclencher.

```
if (Input.textContent !== '') {  
  if (e.code === "Enter"){  
    clearTimeout();  
    minInput = Input.textContent.toLowerCase();  
    checkCountRotate = 0;  
    for (let i = 0; i < guessing.length; i++) { ...  
    }  
    if (checkCountRotate === guessing.length) { ...  
    }  
  }  
  if (e.code === 'Backspace'){  
    Input.textContent = Input.textContent.slice(0, -1);  
    return;  
  }  
}
```

Si le contenu de la réponse n'est pas vide et que la touche pressée est la touche "entrée" alors un certain nombre d'actions auront lieu.

Si un timeout a lieu en arrière-plan, celui-ci sera coupé. Ensuite tout le contenu de la réponse sera transformé en minuscule pour faciliter le traitement et la vérification de la réponse. Une variable de contrôle va être initiée et une boucle vérifiant toutes les réponses possibles va être démarrée.



```

for (let i = 0; i < guessing.length; i++) {
  if (levenshteinDistance(minInput, guessingMin[i]) <= guessing[i].percent) {
    check = checkAlreadyGuess(guessing[i].name);
    if (check === true) {
      overlay.classList.add('none');
      Input.textContent = '';
      return;
    }
    checkScore();
    contentModal(guessing[i].name);
    if (rowThree.classList.contains('none') === true) {
      rowThree.classList.remove('none');
    }
    overlay.classList.add('none');
    modalDesc.classList.remove('none');
    listLanguageFound.innerHTML = listLanguageFound.innerHTML + "<li id='"+ guessing[i].name + "'>" + guessing[i].name + "</li>";
    found.push(guessing[i].name);
    localStorage.setItem('found', JSON.stringify(found));
    if (document.querySelector('#checkBoxClosing').checked === true) {
      setTimeout(() => {
        modalDesc.classList.add('none');
      }, 2000);
    }
  }
  return;
}
checkCountRotate++;

```

La boucle contient deux parties : une condition qui va comparer la distance de Levenstein de la réponse entrée par l'utilisateur avec le résultat maximal autorisé.

Pour faire court et simple, la distance de Levenshtein est une mesure mathématique donnant la différence entre deux chaînes de caractères. Cela inclut l'ajout, le retrait et le remplacement des caractères.

```

const levenshteinDistance = (str1 = '', str2 = '') => {
  const track = Array(str2.length + 1).fill(null).map(() =>
    Array(str1.length + 1).fill(null));
  for (let i = 0; i <= str1.length; i += 1) {
    track[0][i] = i;
  }
  for (let j = 0; j <= str2.length; j += 1) {
    track[j][0] = j;
  }
  for (let j = 1; j <= str2.length; j += 1) {
    for (let i = 1; i <= str1.length; i += 1) {
      const indicator = str1[i - 1] === str2[j - 1] ? 0 : 1;
      track[j][i] = Math.min(
        track[j][i - 1] + 1, // deletion
        track[j - 1][i] + 1, // insertion
        track[j - 1][i - 1] + indicator, // substitution
      );
    }
  }
  return track[str2.length][str1.length];
};

```

Si la condition est valide , c'est à dire que le résultat de la distance de Levenshtein est inférieur ou égale au maximum autorisé, alors une vérification dans le tableau des réponses déjà trouvés sera effectuée. Ensuite une vérification du score sera effectuée , pour déclencher ou non la victoire.

```
function checkScore() {
  if (score === guessing.length - 1) {
    clearTimeout();
    setTimeout(() => {
      result.textContent = 'Vous avez gagné !';
      afterGame.classList.add('fadeIn');
      afterGame.classList.remove('none');
      setTimeout(() => {
        mainGame.classList.add('none');
      }, 1000);
      localStorage.removeItem('found');
      localStorage.removeItem('score');
      localStorage.removeItem('error');
      return;
    }, 5000);
  }
  else{
    Input.textContent = '';
    score++;
    localStorage.setItem('score', score);
    countFound.textContent = score;
  }
}
```

Si la victoire n'est pas déclenchée, alors la modal détaillant un peu plus d'information sur le langage sera rempli via la fonction suivante.

```
function contentModal(target){
  for (let i = 0; i < modalContentArray.length; i++) {
    if (modalContentArray[i].name === target) {
      titleModal.textContent = modalContentArray[i].name;
      languageDesc.textContent = modalContentArray[i].description;
      modalImg.src = 'assets/img/' + modalContentArray[i].img;
      modalImg.alt = modalContentArray[i].alt;
      break;
    }
  }
}
```

Juste après, la modal de description du langage sera ouverte tout en cachant l'overlay de réponses. La liste des langages déjà trouvée lors de cette partie sera alors mise à jour sur le jeu et sera aussi enregistrée dans le stockage local pour sauvegarder en cas de souhait de reprises plus tard. Si la case de fermeture automatique de la fenêtre est cochée , un minuteur de 2 secondes se lancera pour fermer cette fenêtre.

Dans le cas où la boucle de vérification arrive à son terme sans pour autant avoir trouvé de résultats positifs , une condition vérifiera que cette boucle a bien fait le même nombre d'itération que la longueur de la liste de réponses possibles.

```
if (checkCountRotate === guessing.length) {  
    switch(erreur){  
        case 0:  
            Input.textContent = '';  
            erreur++;  
            document.querySelector('.error1').classList.add('lightBlue');  
            localStorage.setItem('error', erreur);  
            break;  
        case 1:  
            Input.textContent = '';  
            erreur++;  
            document.querySelector('.error2').classList.add('lightBlue');  
            localStorage.setItem('error', erreur);  
            break;  
        case 2:  
            Input.textContent = '';  
            erreur++;  
            document.querySelector('.error3').classList.add('lightBlue');  
            result.innerHTML = "Vous avez perdu ! <br> Vous avez trouvé "+score+" langages  
            présents dans l'image";  
            afterGame.classList.add('fadeIn');  
            afterGame.classList.remove('none');  
            setTimeout(() => {  
                mainGame.classList.add('none');  
            }, 1000);  
            break;  
    }  
    overlay.classList.add('none');  
    return;  
}
```

Si la condition valide le fait que la boucle de contrôle précédente n'as pas trouvé de résultat correct, un switch sera déclenché pour vérifier combien d'erreurs l'utilisateur a fait. En fonction de la valeur, celle-ci ajoutera une erreur au compteur ou déclenchera la défaite.

Si la défaite est déclenchée, l'écran final sera rempli avec un message indiquant combien de langages ont été trouvés. L'overlay de réponses sera caché dans un même temps.

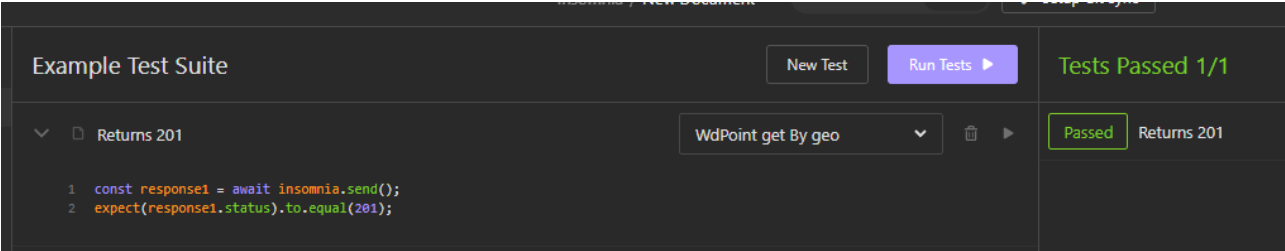
La dernière condition est juste une condition vérifiant si la touche pressée est le retour arrière. Si c'est le cas , le javascript récupérera le contenu de la réponse et effacera le dernier caractère de celle-ci.

```
if (e.code === 'Backspace'){  
    Input.textContent = Input.textContent.slice(0, -1);  
    return;  
}
```

# 5. Jeu d'essai de la fonctionnalité la plus représentative

Pour cette partie , le jeu d'essai portera sur une fonctionnalité expliquée précédemment, le contrôleur de recherche par géolocalisation et par critères spéciaux.

Pour les besoins des tests , l'absence d'une interface front-end n'est pas un souci étant donnée l'existence de plusieurs logiciels dédiés aux tests des API. Pour les phases de tests , nous nous appuierons donc sur Insomnia , logiciel de tests et de développement d'API créé par Kong Inc.



Voici un simple test vérifiant le code renvoyé par l'API. Cela indique juste que l'API a répondu avec le bon code. Mais ce n'est pas suffisant, nous allons donc vérifier directement envoyer une demande au format JSON , contenant tous les paramètres souhaités. Pour de futures références dans cette partie , voici la liste présente des produits actuellement enregistrés ainsi que les différents points de retrait.

## Listes des produits:

	id	category_id	user_id	reduction_id	name	description	detail	price	wd_point_id
<input type="checkbox"/> Edit Copy Delete	1	2	1	2	Test	Velit sit iriure at et sanctus et lorem takimata	Invidunt dolor nostrud et nulla	29.99	2
<input type="checkbox"/> Edit Copy Delete	2	1	1	3	Test	Velit sit iriure at et sanctus et lorem takimata	Invidunt dolor nostrud et nulla	29.99	2
<input type="checkbox"/> Edit Copy Delete	3	1	1	1	Test	Velit sit iriure at et sanctus et lorem takimata	Invidunt dolor nostrud et nulla	29.99	4
<input type="checkbox"/> Edit Copy Delete	4	2	1	1	Test	Velit sit iriure at et sanctus et lorem takimata	Invidunt dolor nostrud et nulla	29.99	5
<input type="checkbox"/> Edit Copy Delete	5	3	1	1	Test	Velit sit iriure at et sanctus et lorem takimata	Invidunt dolor nostrud et nulla	29.99	3

## Liste des points de retraits:

	id	user_id	address	lat	lon	dep
<input type="checkbox"/> Edit Copy Delete	2	2	16 rue tamisier, lons le saunier	46.6747295	5.5524935	39
<input type="checkbox"/> Edit Copy Delete	3	1	20 rue tamisier, lons le saunier	46.674572	5.552366	39
<input type="checkbox"/> Edit Copy Delete	4	2	22 rue du bourg neuf, laons	48.7060899	1.1799159	28
<input type="checkbox"/> Edit Copy Delete	5	1	19 rue de la république, besançon	47.2396692	6.02641	25

Cette fonctionnalité se basant sur les distances à vol d'oiseau depuis le point de départ vers les différents points de retraits, il se peut qu'elle soit par moment faussée , notamment dans un milieu montagneux.

Voici la forme de base des données envoyées. Pour le moment, celle-ci ne contient qu'une adresse et une limite de distance mais à déjà la possibilité d'ajouter des critères de recherches bien spécifiques.

L'adresse pointant sur l'API a été retirée pour parer à tout soucis de sécurité , celle-ci n'étant pas encore opérationnel mais pouvant tout de même donner quelques informations sur la structures des adresse employés pour les appel à l'API .

```
POST https://api.myproduct.com/v1/users
JSON Auth Query Headers 2 Docs
1 {
2   "address": "2 route de montaigu, lons le saunier",
3   "lat": "",
4   "lon": "",
5   "dep": "",
6   "limitKlm": "10",
7   "levelReductionFilter": "",
8   "categoryFilter": "",
9   "priceMinFilter": "",
10  "priceMaxFilter": "",
11  "orderFilter": ""
12 }
```

Pour le moment , voici ce que le logiciel nous retourne lorsque la requête est émise.

```
{
  "@context": "\/api\/contexts\/WdPoint",
  "@id": "\/api\/wd_points",
  "@type": "hydra:Collection",
  "hydra:member": [
    {
      "id produit": 2,
      "nom produit": "Test",
      "adresse du point de retrait": "16 rue tamisier, lons le saunier"
    },
    {
      "id produit": 1,
      "nom produit": "Test",
      "adresse du point de retrait": "16 rue tamisier, lons le saunier"
    },
    {
      "id produit": 5,
      "nom produit": "Test",
      "adresse du point de retrait": "20 rue tamisier, lons le saunier"
    }
  ],
  "success1": true,
  "hydra:totalItems": 2
}
```

Deux produits n'apparaissent pas car leur point de retrait respectif se trouve en dehors de la limite kilométrique définie par l'utilisateur.

Pour les besoins de cette partie et montré que cette fonctionnalité est en état de fonctionnement total, un deuxième test va être réalisé , cette fois-ci en ajoutant plusieurs critères de recherches de façon à ne cibler qu'un seul et unique produit.

Les paramètres sont le niveau de réduction ( chaque niveau correspondant à un pourcentage enregistré en base de données) et la catégorie du produit.

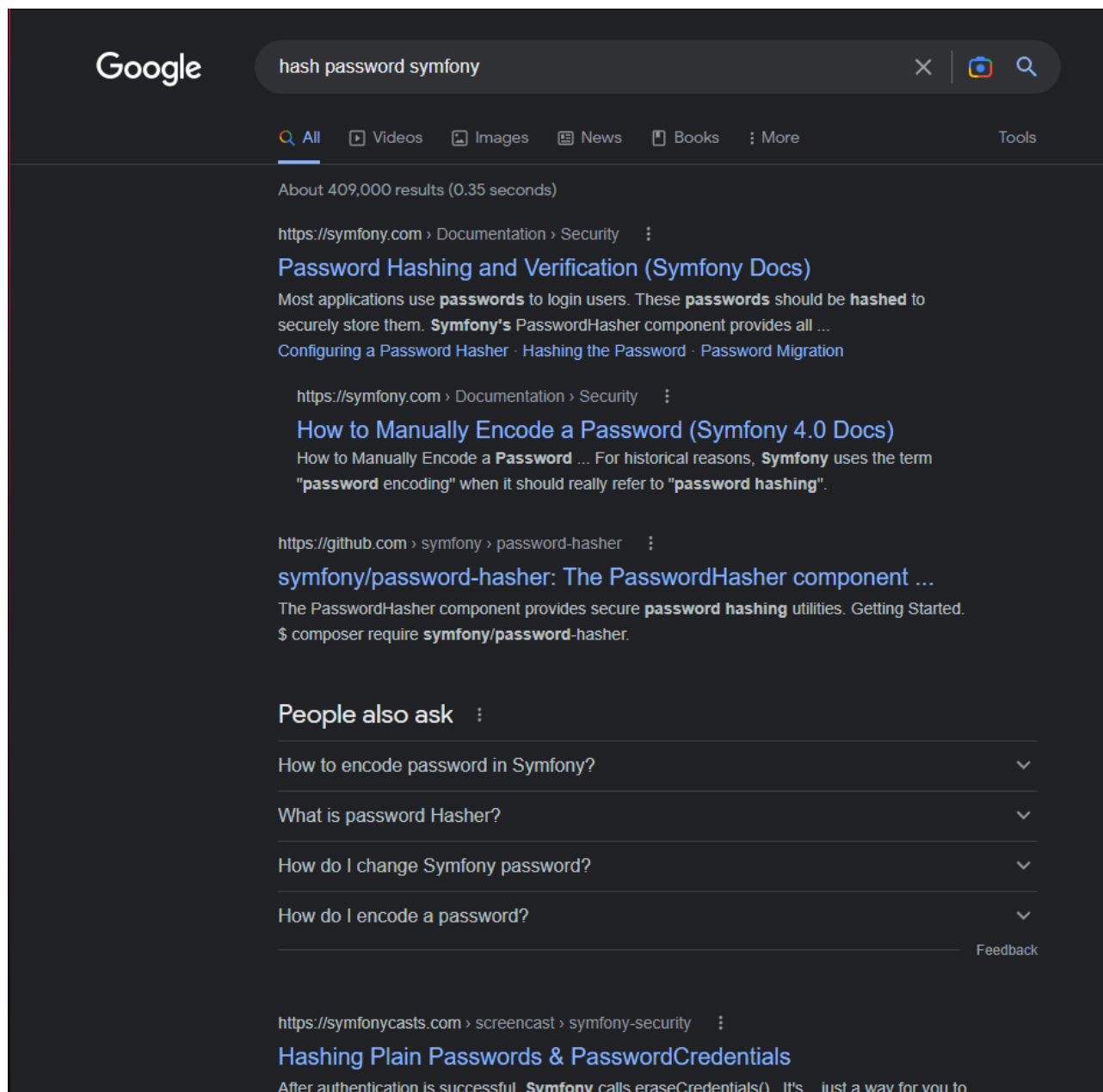
```
{
  "address": "2 route de montaigu, lons le saunier",
  "lat": "",
  "lon": "",
  "dep": "",
  "limitKlm": "10",
  "levelReductionFilter": "2",
  "categoryFilter": "2",
  "priceMinFilter": "",
  "priceMaxFilter": "",
  "orderFilter": ""
}
```

Voici le résultat donné par Insomnia , un seul et unique produit qui correspond aux critères demandés si on l'observe dans la liste des produits enregistrés sur phpMyAdmin précédemment cités.

```
{
  "@context": "\/api\/contexts\/wdPoint",
  "@id": "\/api\/wd_points",
  "@type": "hydra:Collection",
  "hydra:member": [
    [
      {
        "id produit": 1,
        "nom produit": "Test",
        "adresse du point de retrait": "16 rue tamisier, lons le saunier"
      }
    ],
    "success1"
  ],
  "hydra:totalItems": 2
}
```

## 6. Description de la veille de sécurité

Sur la question de la sécurité, j'ai effectué ma veille sur la documentation officielle de Symfony qui décrit très bien les principes d'authentification, de vérification de l'utilisateur ainsi que d'autorisation d'accès. Dans la plupart des cas, j'écrivais des mots clés liés à mon problème et le premier résultat était souvent la documentation officielle. Par exemple « hash password symfony »



Je vais donc dans cette partie démontrer tout ce que j'ai utilisé pour sécuriser au mieux l'application.

Dans un premier temps, du côté de l'utilisateur j'ai montré comment bloquer le back-office aux utilisateurs qui ne disposent pas du bon rôle. Pour ce qui est de la connexion, Symfony a une façon presque magique de la gérer. En effet, via la configuration de ses « providers » et « firewall » depuis le fichier `security.yaml`. Le « provider » est en fait l'utilisateur courant. Ainsi pour accéder aux parties sécurisées de l'application, je peux définir où elle ira chercher cette instance de provider et à partir de quelle donnée qui lui est liée. Ici, je décide que le provider viendra de ma base de données et plus précisément de l'entité User. Pour être encore plus précis, la propriété à vérifier sera l'email.

Le « Firewall » prend en charge chaque requête et vérifie si elle nécessite un l'authentification d'un utilisateur, auquel cas il demandera l'authentification pour y accéder. Ici, je défini le provider de mon firewall, puis je lui défini le nom de la route de déconnexion, et enfin celle de connexion. Le système de connexion est une chose, mais il faut sécuriser les données des utilisateurs, notamment pour tout ce qui est relatif à cette connexion.

Je retourne dans le fichier `security.yaml` pour y ajouter la ligne `password_hasher` et définir un algorithme de hashage pour mon entité User dont je spécifie le namespace. Ici, je règle l'algorithme sur « auto » afin que soit perpétuellement défini l'algorithme le plus performant. Hasher un mot de passe c'est le traduire en une série de caractère qui le masque totalement grâce à un algorithme, c'est une méthode de protection de mot de passe basique. De fait, Symfony propose un service natif pour hasher les mots de passe. De retour dans mes fixtures, il me suffit d'instancier ce service et d'appeler la méthode `hashPassword()` sur les mots de passe à protéger.



## 7. Situation de travail ayant nécessité une recherche en anglais

Tout au long de mon stage, et de manière générale des 12 derniers mois, j'ai effectué toutes mes recherches en anglais. Que ce soit pour le nombre de résultats ou pour la pertinence de ceux-ci, la langue de Shakespeare donnera toujours de meilleurs résultats en développement, sans oublier que cela fera très souvent ressortir StackOverflow ( forum d'entraides entre développeurs ) dans les premiers résultats.

Le cas le plus pertinent que je pourrais citer pour ces recherches serait lors de ma recherche d'API de géocodage. Toutes les API trouvées avait une documentation principalement rédigé en anglais, la recherche en anglais donnait donc plus de résultats.

## 8. Extrait d'un site utilisé lors des recherches en anglais accompagné d'une traduction.

source : <https://symfony.com/doc/5.4/doctrine.html#migrations-creating-the-database-tables-schema>

### English:

The `make:entity` command is a tool to make life easier. But this is your code: add/remove fields, add/remove methods or update configuration.

Doctrine supports a wide variety of field types, each with their own options. To see a full list, check out [Doctrine's Mapping Types documentation](#). If you want to use XML instead of annotations, add `type: xml` and `dir: '%kernel.project_dir%/config/doctrine'` to the entity mappings in your `config/packages/doctrine.yaml` file.

Be careful not to use reserved SQL keywords as your table or column names (e.g. GROUP or USER). See Doctrine's [Reserved SQL keywords documentation](#) for details on how to escape these. Or, change the table name with `#[ORM\Table(name: "groups")]` above the class or configure the column name with the `name: "group_name"` option.

### [Migrations: Creating the Database Tables/Schema](#)

The Product class is fully-configured and ready to save to a product table. If you just defined this class, your database doesn't actually have the product table yet. To add it, you can leverage the [DoctrineMigrationsBundle](#), which is already installed:

## Français:

La commande “ make:entity ” est un outil qui simplifie la vie. Mais voilà votre code: ajouter/supprimer des champs, ajouter/supprimer des méthodes ou mettre à jour des configurations.

Doctrine supporte une large variété de types de champs, chacun avec ses propres options. Pour voir une liste complète , consultez la documentation des types de mapping de Doctrine. Si vous souhaitez utiliser un format XML plutôt que des annotations , ajouter: “ type:xml ” et “ dir: '%kernel.project\_dir%/config/doctrine' ” à l'entité mappings dans le fichier “ config/packages/doctrine.yaml ”.

Faites attention à ne pas utiliser de mot-clés réservé au SQL dans vos noms de tables ou de colonnes ( par exemple GROUP ou USER). Consulter la documentation des mots-clés réservés au SQL de Doctrine pour plus de détails sur la façon de les échapper. Ou, changez le nom de la table avec “ #[ORM\Table(name: “groups”)] ” au-dessus de la classe ou configurez le nom de la colonne avec l'option : “group\_name”.

## [Migration: Créer les tables et le schéma de la base de donnée](#)

La classe Product est entièrement configurée et prête à être enregistrée en tant que table Product. Si vous n'avez que défini cette classe , la base de donnée ne possède actuellement pas la table associée. Pour l'ajouter , vous pouvez utiliser le DoctrineMigrationBundle, qui est déjà installé.

## 9. Conclusion

Pour cette fin de dossier , j'aimerais faire un point sur les apports de ce stage.

Tout d'abord, ce stage a essentiellement couvert du développement Back-End. Cela m'a permis de vraiment voir ce que je préférais dans le développement web. Bien que j'aime toute la logique nécessaire au Back-End et n'apprécie que très peu tout ce qui est visuel, faire autant de PHP et utiliser des frameworks php m'as fait prendre conscience que ce que j'apprécie vraiment se trouve être la liberté du Javascript.

Cependant je n'ai pas non plus passé un mauvais moment, j'ai clairement apprécié apprendre à me servir de Symfony , à poser ma réflexion pour ne réfléchir qu'à un accès distant et une utilisation distante des données.

Ensuite, ma capacité de travail en équipe s'en est grandement améliorée. La disponibilité de Loïc ainsi que celle d'Hamza ( Autre stagiaire) m'ont permis d'améliorer ma communication avec les autres , bien qu'il y ai encore du travail.

Pour conclure cette partie et ce dossier, ce stage n'as fait que renforcer ma volonté à continuer dans le développement Web tout en m'aidant à voir quel langage j'appréciais vraiment.