

Dédicaces

« Celui qui ne remercie pas les gens ne remercie pas Allah. ».

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

Je remercie infiniment mes encadrants technique **TABTI ABDELHAK** et **KHADIRI ISSAM** pédagogique et professionnel pour leur disponibilité, leurs remarques constructives, ses précieux conseils et directives durant toute la période de l'établissement de ce travail.

A nos très chers parents :

Qui nous a soutenus par leurs amours et leurs efforts.

Qui nous ont toujours encouragés.

Pendant toute la période de nos études et qui n'ont épargné aucun effort pour répondre à nos exigences.

Nous espérons être à la hauteur de l'image qu'ils se sont fait de nous tous.

A tous nos amis et amies. A tous nos collègues à l'EHEI.

A nos professeurs.

Pour tout le soutien que vous nous avez apporté. On vous dit merci.

A tous ceux que nous aimons et qui nous aiment, nous dédions ce travail.

Résumé

Ce rapport est la synthèse du travail effectué dans le cadre de mon projet de fin d'études chez SQLI, réalisé pour le compte de Tarkett. Le projet visait à migrer le composant Tarkett Locator, développé initialement sur Symfony 5.3, API Platform 2.6 et PHP 7.3, vers des versions stable et récentes : Symfony 6.4, API Platform 3.2 et PHP 8.3.

La migration avait pour objectifs principaux d'améliorer la sécurité, la performance et la maintenabilité de l'application. Mon travail a commencé par un audit des dépendances pour identifier les vulnérabilités de sécurité et les composants obsolètes. Ensuite, j'ai effectué des mises à jour incrémentales de chaque technologie, en résolvant les conflits et en adaptant le code existant aux nouvelles versions. Chaque étape de la migration a été suivie de tests rigoureux pour s'assurer de la stabilité et du bon fonctionnement de l'application.

Les résultats obtenus montrent une nette amélioration des performances de l'application, une meilleure sécurité grâce à la correction des vulnérabilités identifiées et une facilitation de la maintenabilité du code. Ce projet a permis de moderniser un composant critique pour Tarkett, tout en démontrant l'importance des mises à jour régulières et de l'adaptation aux nouvelles technologies dans le développement logiciel.

Ce rapport conclut que la migration a été un succès, atteignant tous les objectifs fixés, et souligne les compétences techniques et méthodologiques acquises au cours de ce projet.

Abstract

This report summarizes the work done as part of my end-of-study project at SQLI, carried out for the client Tarkett. The project aimed to migrate the Tarkett Locator component, initially developed on Symfony 5.3, API Platform 2.6, and PHP 7.3, to recent and stable versions: Symfony 6.4, API Platform 3.2, and PHP 8.3.

The main objectives of the migration were to improve the application's security, performance, and maintainability. My work began with an audit of the dependencies to identify security vulnerabilities and outdated components. I then performed incremental updates for each technology, resolving conflicts and adapting the existing code to the new versions. Each step of the migration was followed by thorough testing to ensure the application's stability and proper functioning.

The results show a significant improvement in the application's performance, enhanced security through the resolution of identified vulnerabilities, and easier code maintainability. This project modernized a critical component for Tarkett while demonstrating the importance of regular updates and adaptation to new technologies in software development.

This report concludes that the migration was successful, achieving all the set objectives, and highlights the technical and methodological skills acquired during this project.

Table des figures

Figure 1: Logo SQLI Digital Expérience	14
Figure 2: Répartition des agences.....	15
Figure 3: Partenaires de SQLI	17
Figure 4: Logo Tarkett	19
Figure 5: Tarkett dans le monde	19
Figure 5: Répartition du capital	20
Figure 5: Le flux de données de Tarkett	24
Figure 6: Processus Scrum	29
Figure 8: Processus Microsoft Teams	32
Figure 9: Les vulnérabilités de sécurité et les dépendances abandonné.....	37
Figure 10: Les vulnérabilités de sécurité	38
Figure 11: Détails de vulnérabilité de sécurité d'API Platform	39
Figure 12: API Platform packagist.org	40
Figure 13: Le fichier de configuration security.yaml	40
Figure 14: La version du PHP	41
Figure 15: Le fichier de configuration api_platform.yaml	41
Figure 16: l'ancien Api resource	42
Figure 16: l'ancien annotation.....	42
Figure 17: le nouveau Api resource	43
Figure 18: le nouveau attribute.....	43
Figure 19: les espaces de nom	43
Figure 20: le fichier de configuration de Doctrine	44
Figure 21: la déclaration d'un variable	45
Figure 22: Data persister.....	45
Figure 24: conflit du version PHP.....	47
Figure 25: test OK.....	48
Figure 26: les espaces de nom	49
Figure 27 : le fichier de configuration d'API Platform	49
Figure 28: les résultats du test	50
Figure 29: la version du PHP.....	51
Figure 30: Logo PHP	53
Figure 31: Logo Symfony	54
Figure 32: Logo PHPUnit	54
Figure 33: Logo Git	55
Figure 34: Logo Github.....	55

Figure 35: Logo Docker	56
Figure 36 : logo Composer	56
Figure 37: Logo PhpStorm	57
Figure 38: Logo MySQL	57
Figure 39 : l'architecture du composant Tarkett Locator	61
Figure 40: la version du Symfony	61
Figure 41: les tests OK	62
Figure 42: la page web du Tarkett Locator	62
Figure 43: la version d'API Platform	63
Figure 44: la page web du Swagger	63
Figure 45: la version du PHP	63

Table de matières

Dédicaces.....	3
Résumé.....	4
Abstract	5
Table des figures.....	6
Table de matières	8
Liste des abréviations	10
INTRODUCTION GÉNÉRALE	11
1.1. Introduction	14
1.2. Organisme d'accueil.....	14
1.2.1 Présentation du Groupe SQLI	14
1.3. Contexte du projet « Tarkett »	18
1.3.1.Stratégie du « tarkett »	20
<i>impacT 2027</i> se concentre sur 4 piliers	20
1.3.2. Présentation du projet « Tarkett Locator ».....	23
1.3.3. Objectifs de Tarkett	21
1.3.4. Description de l'environnement du projet « tarkett locator »	23
Collecte des Données	24
1.4.1. Etude de l'existant.....	26
1.4.2. Solution Proposée par le projet	27
1.5. Planification et Conduite du projet	27
1.5.1. Méthodologie SCRUM.....	28
1.5.2. Les Outils de communication	32
1.5.3. Conclusion	32
CHAPITRE 2 : ETUDE FONCTIONNELLE	33
2.1. Introduction	34
2.2 Vérification des dépendances obsolètes et des vulnérabilités de sécurité (Symfony 5.3)	34

2.2.1 Introduction	34
2.2.2 Utilisation de composer audit	34
2.3 la migration du Symfony 5.3 vers 5.4:	37
2.4 la migration du PHP 7.4 vers 8.1:	41
2.5 Vérification avec des Tests Unitaires	46
2.6 la migration du Symfony 5.4 vers 6.4:	46
2.6.1 la migration d'API Platform du 2.7 vers 3.0:	48
2.6.2 la migration d'API Platform du 3.0 vers 3.2:	50
2.7 la migration du PHP 8.1 vers 8.3:	50
2.7 Conclusion :	51
CHAPITRE 3 : TECHNOLOGIES ET OUTILS UTILISES	52
3.1. Introduction.....	53
3.2. Technologies et Outils de développement	53
3.2.1. Langages de programmation.....	53
3.2.2. Bibliothèques et Frameworks.....	54
3.2.3. Outils de développement.....	55
3.2.4. Système de gestion de la base de données	57
3.3. Conclusion	58
CHAPITRE 4 : REALISATION DU PROJET	59
4.1. Introduction	60
4.2 Migration du Framework Symfony.....	60
4.3 Migration d'API Platform	63
4.4 Migration du PHP	63
4.4. Conclusion	64

Liste des abréviations

Abréviation	Désignation
API	Application Programming Language
PHP	Hypertext Preprocessor
ORM	Object-Relational Mapping
SQL	Structured Query Language
B2B	Business to business
B2C	Business to customer

INTRODUCTION GÉNÉRALE

Les nouvelles technologies et les mises à jour logicielles jouent un rôle crucial dans le maintien et l'amélioration des systèmes informatiques. Dans un environnement où les exigences des utilisateurs et les normes de sécurité évoluent rapidement, il est essentiel pour les entreprises de mettre à jour régulièrement leurs infrastructures technologiques pour rester compétitives et sécurisées.

Dans le cadre de mon stage chez SQLI, j'ai eu l'opportunité de travailler avec le client Tarkett, leader mondial dans la fabrication et la vente de revêtements de sol. Un de leurs composants critiques, le Tarkett Locator, permet aux clients et partenaires de localiser facilement les points de vente et les distributeurs des produits Tarkett. Initialement développé sur Symfony 5.3, API Platform 2.6 et PHP 7.3, ce système est devenu obsolète, présentant des vulnérabilités de sécurité et des limitations de performance.

L'objectif principal de ce projet de fin d'études est de migrer le Tarkett Locator vers des versions plus récentes et robustes de ses technologies sous-jacentes : Symfony 6.4, API Platform 3.2, et PHP 8.3. Cette migration vise à améliorer la sécurité, la performance, et la maintenabilité de l'application, tout en assurant la continuité des services offerts aux utilisateurs.

La réalisation de ce projet s'inscrit dans une démarche méthodique et structurée. Elle comprend une analyse préalable des dépendances et des vulnérabilités, suivie d'une migration incrémentale et d'une série de tests rigoureux pour garantir la stabilité et la fiabilité de l'application. Cette approche permet de minimiser les risques associés aux mises à jour majeures et d'assurer une transition fluide et efficace vers les nouvelles versions.

Ce rapport détaille les étapes de la migration, les défis rencontrés et les solutions apportées, ainsi que les bénéfices observés après la mise à jour. Il constitue une synthèse du travail effectué et des compétences acquises au cours de ce projet, en soulignant l'importance de l'innovation technologique et de la gestion proactive des systèmes d'information dans un contexte professionnel.

En conclusion, ce projet de fin d'études représente une contribution significative à l'amélioration continue de l'infrastructure technologique de Tarkett, démontrant l'impact positif des mises à jour logicielles sur la performance et la sécurité des applications critiques pour l'entreprise.

CHAPITRE 1 : CONTEXTE DU PROJET

1.1. Introduction

Ce premier chapitre est pour mieux circonscrire le contexte de déroulement du stage, une présentation de l'organisme d'accueil SQLI GROUPE et du client GROUPE TARKETT s'impose, par la suite nous allons définir les tâches, les objectifs de ce travail, l'équipe impliqué dans ce travail pour finir avec une description de contenu de ce rapport.

1.2. Organisme d'accueil

1.2.1 Présentation du Groupe SQLI

SQLI est un groupe européen de services dédié au monde du Digital, spécialisé dans la conception, la mise en œuvre, le déploiement mondial et l'exploitation de dispositifs omnicanal. Son positionnement de spécialiste du commerce et des technologies permet à ses équipes d'experts d'accompagner durablement les grandes entreprises et marques européennes dans le développement de leurs ventes et de leur notoriété ainsi que leur performance interne en réinventant l'expérience client, partenaire et collaborateur.



Figure 1: Logo SQLI Digital Expérience

Elle est organisée en agences de proximité, afin de conserver le maximum de réactivité face aux besoins de ses clients. Son positionnement unique lui permet de répondre de façon globale aux enjeux de performance :

- Performance business : e-commerce, marketing digital & social, mobilité e-communication
- ...
- Performance de l'entreprise : entreprise collaborative, poste de travail, solutions métier, intégration de S.I ...

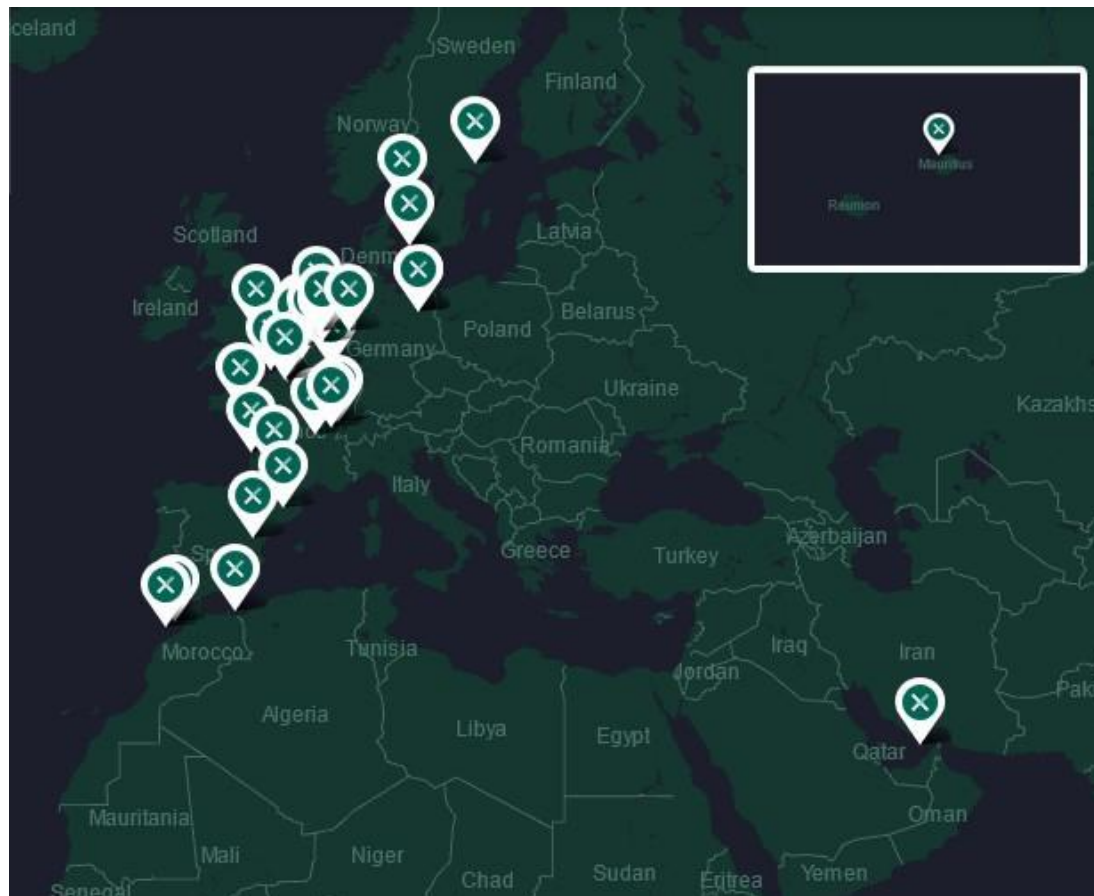


Figure 2: Répartition des agences

Ses +2400 collaborateurs sont répartis en France (Paris, Lyon, Toulouse, Bordeaux, Rouen et Nantes), en Suisse (Lausanne et Genève), au Luxembourg, en Belgique (Bruxelles), aux Pays-Bas et au Maroc (Oujda, Rabat et Casablanca)

Le groupe SQLI suit un rythme de développement élevé au niveau des projets, des ressources humaines, et chiffre d'affaires :

- 23 ans de projets et missions de haut niveau : SQLI assoit son développement sur une expertise technologique de pointe et sur sa politique intense de veille et R&D. En 2005, SQLI devient la 1ère SSII française à obtenir la certification CMMI niveau 3
- 26 partenaires, leaders des solutions technologiques : Le positionnement de SQLI à la conjonction du monde du digital et du système d'information de l'entreprise est sa valeur ajoutée. Le groupe a noué un réseau de partenaires pour adresser ses défis de façon la plus performante possible, et fournir le meilleur conseil, la meilleure solution

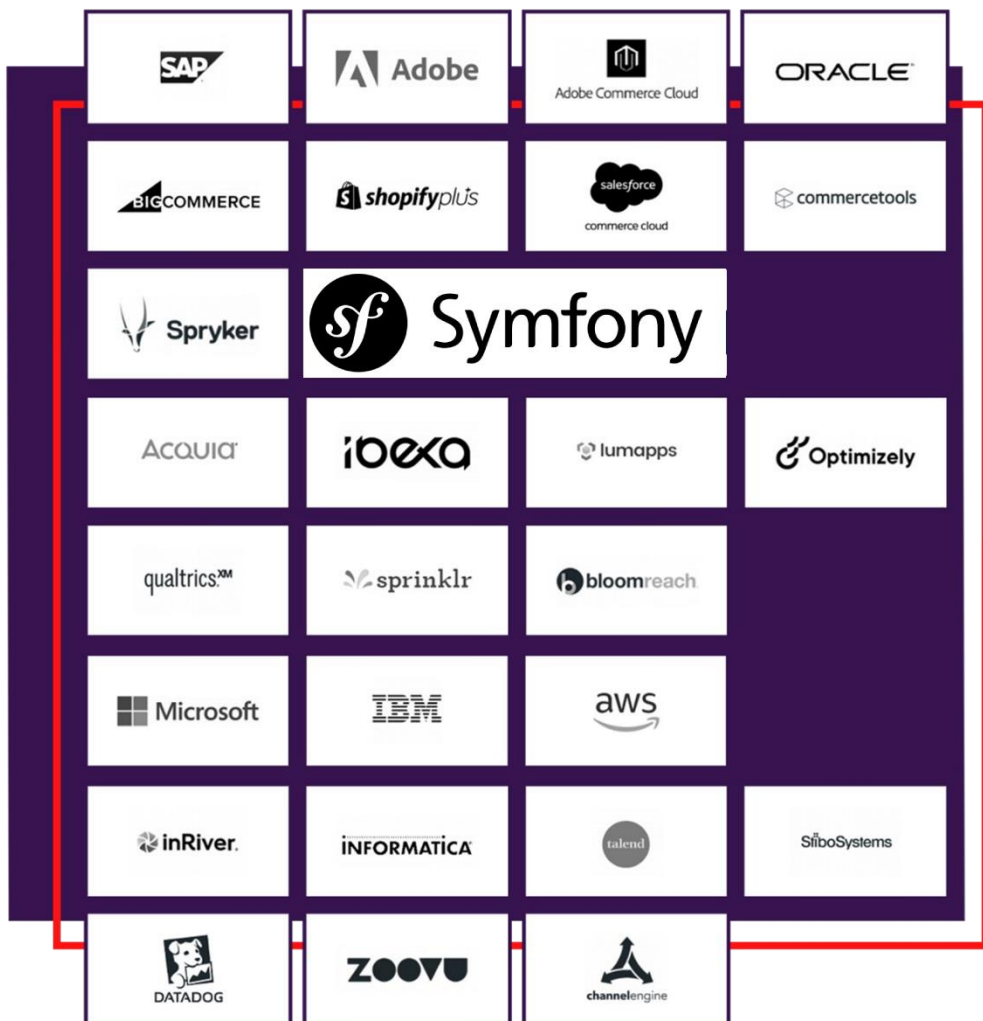


Figure 3: Partenaires de SQLI

1.2.1.1. Expertises clés du Groupe SQLI

- Conseil, Gouvernance, Stratégie digitale, Conduite du changement, Lab Innovation, Agilisation.
- Expérience design, Marketing digital, Stratégie d'acquisition et fidélisation, Web Analytics, Social media.
- E-commerce, Commerce unifié, Commerce Omnicanal,

- Digital in store, PIM et DAM.
- Solutions Web et Mobiles.
- Digital Workplace, Management de l'innovation.
- Digitalisation des processus, Solutions Achats, Finance, RH, Business Intelligence, RGPD.
- Big Data, Mobilité, Objets connectés, Maintenance en condition opérationnelle.
- Socle digital, Stratégie SI, APIisation.

1.2.1.2. SQLI Oujda

Dans le cadre de sa stratégie de développement alliant taille et densité technologique, le Groupe SQLI s'est doté de pôles spécialisés, disposant chacun d'une expertise technologique et/ou métier spécifique. Pour accompagner le développement de son pôle dédié Open Source de Poitiers, en septembre 2006, le Groupe SQLI ouvrait, en partenariat avec l'Université Mohammed Premier, le tout premier centre de R&D / offshore entièrement dédié aux technologies Open Source. Depuis, SQLI bénéficie, au sein même de l'Université, d'un fort potentiel de ressources de très haut niveau de qualité et de locaux entièrement équipés des technologies. En contrepartie, les experts du Groupe interviennent dans le cadre de formations spécifiques, telles que CMMI, et contribuent à l'enrichissement des cursus scolaires de modules complémentaires. Concernant l'organisation du groupe SQLI Oujda, il s'agit d'une répartition selon 3 segments qui à leur tour sont répartis en sept équipes :

- Les technologies et usages internet.
- SAP (Net Weaver, CRM...).
- La Business intelligence (BO, Open Source ...)

1.3. Contexte du projet « Tarkett »



Figure 4: Logo Tarkett

Tarkett est un leader mondial des revêtements de sol et des surfaces sportives. Fort de plus de 140 ans d'histoire, 12 000 collaborateurs et 34 sites de production à travers le monde, cette multinationale française vend plus d'1,3 million de mètres carrés de revêtement de sols par jour. Présent dans plus de 100 pays, le groupe combine la puissance de son réseau mondial à l'expertise de ses équipes locales pour concevoir et produire des gammes de produits et services répondant aux exigences et aux demandes spécifiques de chacun de ses clients.

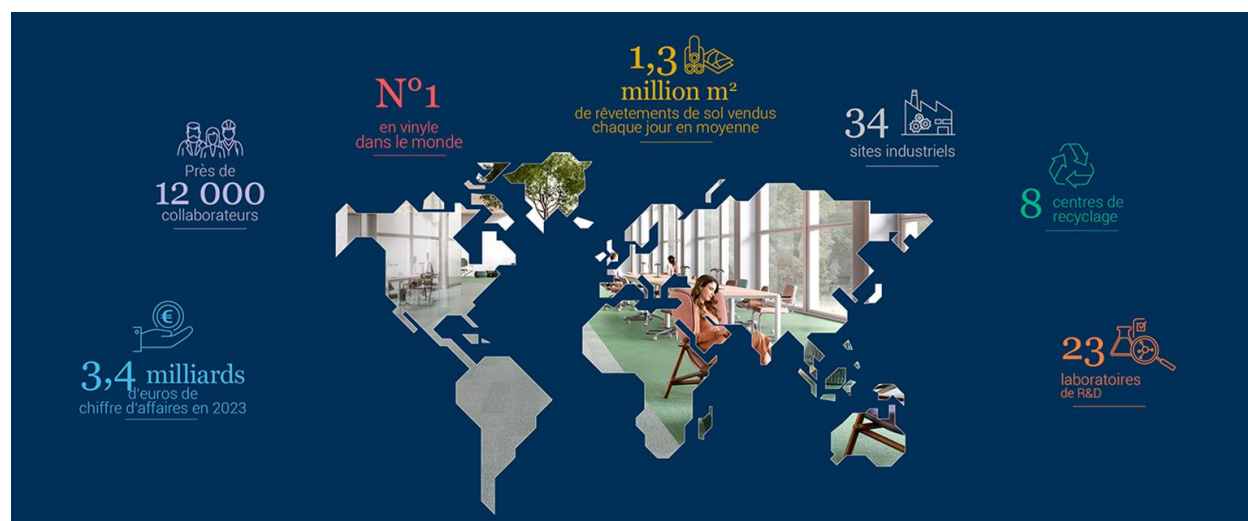


Figure 5: Tarkett dans le monde



Figure 5: Tarkett dans le monde

1.3.1. Stratégie du « tarkett »

Le nouveau plan stratégique de Tarkett, "impacT 2027", est conçu avec une ambition claire : être la société de revêtements de sol et de surfaces sportives la plus innovante, la plus durable, et offrant la meilleure expérience pour nos clients et collaborateurs.

impacT 2027 se concentre sur 4 piliers

- Offrir à nos clients une expérience de premier ordre
- Créer des produits et services innovants
- Être le leader en matière de développement durable
- Donner à nos équipes performantes les moyens de tenir notre promesse

Problématiques Identifiées

2. **Vulnérabilités de Sécurité** : Les versions obsolètes des technologies utilisées présentaient des vulnérabilités connues, exposant l'application à des risques de sécurité.
3. **Performance** : L'application souffrait de limitations de performance, ralentissant les temps de réponse et impactant l'expérience utilisateur.
4. **Maintenabilité** : Le code base étant développé sur des versions non supportées, il devenait de plus en plus difficile de maintenir et de faire évoluer l'application.

1.3.2. Objectifs de Tarkett

Les objectifs du projet « Tarkett » vont au-delà des aspects techniques et incluent plusieurs ambitions stratégiques et commerciales essentielles pour Tarkett. Voici les principaux objectifs non techniques :

Amélioration de l'Expérience Client

Accessibilité Accrue : Faciliter l'accès aux informations sur les points de vente et les distributeurs pour les clients, rendant la recherche rapide et intuitive.

Satisfaction Client : Offrir une expérience utilisateur optimale pour augmenter la satisfaction et fidéliser la clientèle.

Personnalisation du Service : Permettre aux clients de trouver facilement les produits spécifiques qu'ils recherchent, augmentant ainsi leur engagement et leur satisfaction.

Expansion Commerciale

Support à la Vente : Aider les équipes de vente et les distributeurs à fournir des informations précises et à jour aux clients, renforçant ainsi le processus de vente.

Augmentation des Ventes : En facilitant l'accès aux points de vente et en améliorant la visibilité des produits Tarkett, l'outil vise à stimuler les ventes et à accroître les revenus.

Promotion des Produits : Utiliser l'outil pour promouvoir de nouveaux produits et offres spéciales, attirant ainsi davantage de clients vers les points de vente.

Renforcement de la Marque

Image de Marque : Renforcer l'image de marque de Tarkett en offrant un outil moderne et efficace qui démontre l'engagement de l'entreprise envers l'innovation et la satisfaction client.

Présence Digitale : Améliorer la présence digitale de Tarkett, rendant l'entreprise plus visible et accessible en ligne.

Efficacité Opérationnelle

Optimisation des Ressources : Réduire le temps et les efforts nécessaires pour gérer et mettre à jour les informations sur les points de vente, permettant aux équipes de se concentrer sur des tâches à plus forte valeur ajoutée.

Centralisation des Données : Centraliser les informations sur les points de vente dans un seul outil, facilitant la gestion et la mise à jour des données.

Relation avec les Partenaires

Support aux Distributeurs : Fournir aux distributeurs un outil puissant pour attirer et servir les clients, améliorant ainsi la collaboration et la satisfaction des partenaires commerciaux.

Coordination Facilitée : Faciliter la coordination entre les différentes parties prenantes (équipes internes, partenaires, distributeurs) grâce à un accès simplifié et centralisé aux informations.

Développement Durable

Réduction de l'Impact Environnemental : En encourageant les clients à trouver les points de vente les plus proches, réduire les déplacements inutiles et contribuer à un impact environnemental moindre.

1.3.3. Exploration des Modèles B2B et B2C chez Tarkett

Tarkett et le B2B

Pour Tarkett, le B2B représente une part significative de leur activité. Voici comment cela se manifeste :

Clients : Tarkett fournit des produits à des entreprises dans divers secteurs, y compris les constructeurs, les architectes, les designers d'intérieur, les institutions publiques, les établissements de santé, les écoles, et les centres sportifs.

Tarkett et le B2C

Bien que Tarkett soit fortement orienté vers le B2B, le B2C joue également un rôle, surtout dans le segment résidentiel :

Clients : Les consommateurs individuels à la recherche de solutions de revêtements de sol pour leur maison, comme des parquets, des sols en vinyle ou en linoléum.

Distribution : Les produits Tarkett sont disponibles dans les magasins de bricolage, les grandes surfaces spécialisées, et les détaillants en ligne, permettant aux consommateurs

d'accéder facilement à leurs produits.

Exemple de Scénarios B2B et B2C chez Tarkett

B2B : Un architecte travaillant sur un nouveau complexe de bureaux pourrait choisir les solutions de sol Tarkett pour leur durabilité et leur design, et collaborer avec Tarkett pour une installation sur mesure.

B2C : Un particulier rénovant sa maison pourrait acheter des lames de vinyle Tarkett dans un magasin de bricolage pour leur facilité d'installation et leur esthétique moderne. En combinant ces deux approches, Tarkett parvient à couvrir une large part du marché des revêtements de sol, répondant aux besoins des entreprises et des consommateurs individuels.

1.3.4. Présentation du projet « Tarkett Locator »

Le projet Tarkett Locator est un outil essentiel qui permet aux clients et partenaires de localiser facilement les points de vente et les distributeurs des produits Tarkett à travers le monde. Initialement développé sur des versions antérieures de Symfony, API Platform et PHP, ce composant présentait plusieurs limitations, notamment en termes de sécurité, de performance et de maintenabilité.

1.3.5. Description de l'environnement du projet « Tarkett »

Le projet « Tarkett Locator » évolue dans un environnement technologique avancé et interconnecté, nécessitant une infrastructure robuste et performante.

1.3.5.1. Le flux de données

Le flux de données dans le projet « Tarkett » est essentiel pour assurer le bon fonctionnement et l'efficacité de l'application. Voici une description détaillée du flux de données en se basant sur le schéma fourni :

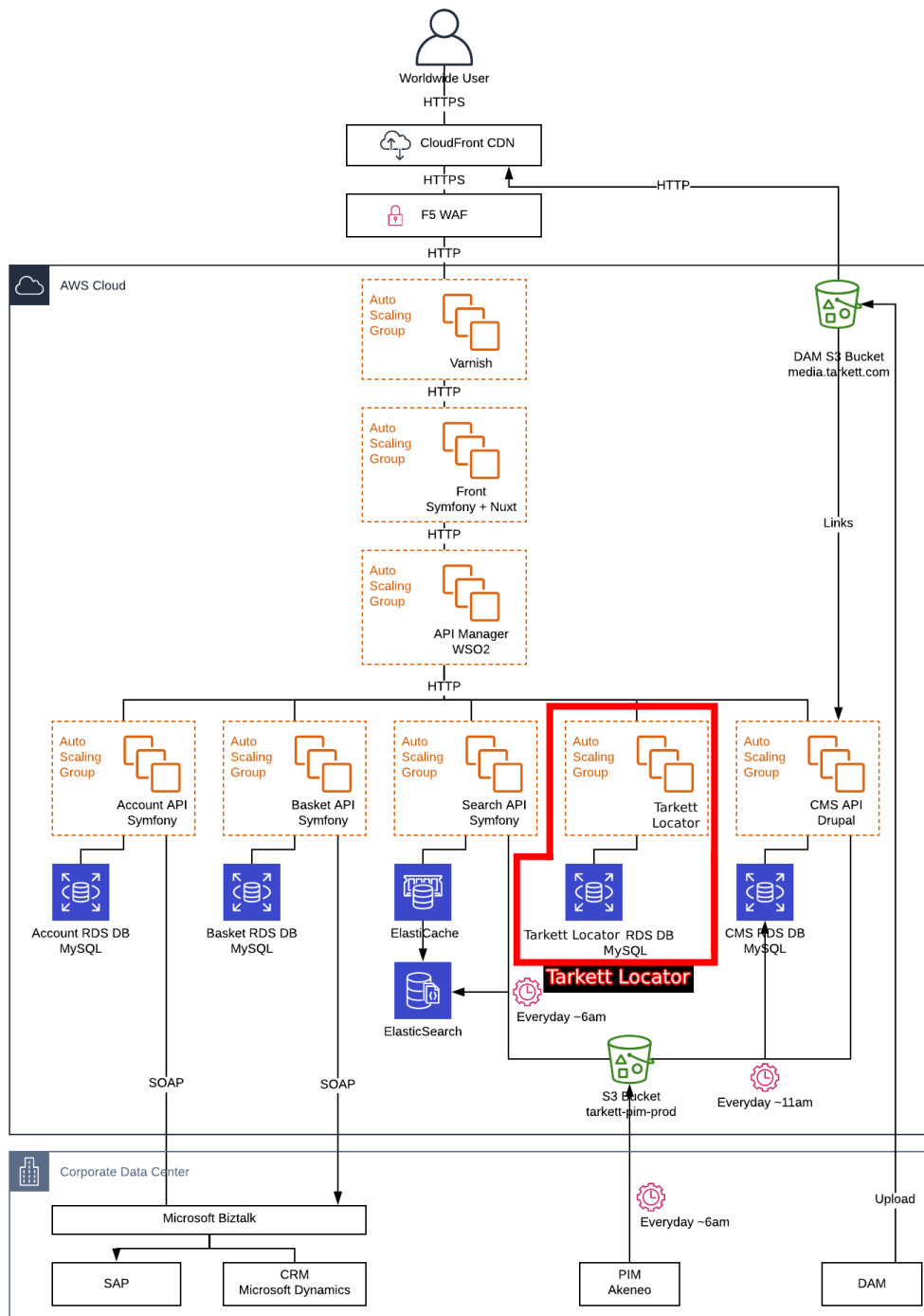


Figure 5: Le flux de données de Tarkett

Collecte des Données

Les données nécessaires pour le Tarkett proviennent de diverses sources :

SAP : Système de gestion intégré fournissant des informations critiques sur les opérations de l'entreprise.

Microsoft Dynamics CRM : Gestion des relations clients, fournissant des données sur les clients et les interactions.

PIM Akeneo : Gestion des informations produits, fournissant des données détaillées sur les produits.

DAM : Système de gestion des actifs numériques, stockant les médias liés aux produits.

Ces données sont collectées via Microsoft Biztalk, qui facilite l'intégration et la communication entre différents systèmes au sein du centre de données d'entreprise.

Traitement des Données

Les données sont également indexées et mises en cache pour optimiser les performances de recherche et de récupération :

ElasticSearch : Utilisé pour l'indexation et la recherche rapide de données.

ElasticCache : Utilisé pour la mise en cache des données fréquemment consultées, réduisant ainsi les temps de réponse.

Distribution des Données

Les données traitées et stockées sont distribuées aux différentes composantes de l'application via des APIs et des services de gestion :

Account API, Basket API, Search API (Symfony) : APIs Symfony utilisées pour gérer les comptes utilisateurs, les paniers d'achat, et les recherches.

API Manager (WSO2) : Gestionnaire d'APIs utilisé pour orchestrer et sécuriser les appels API.

CMS API (Drupal) : API Drupal utilisée pour gérer le contenu du site web.

Ces APIs et services sont déployés dans des groupes de mise à l'échelle automatique (Auto Scaling Groups) sur le cloud AWS pour assurer la disponibilité et la performance.

Affichage des Données

Les données sont affichées aux utilisateurs via une interface utilisateur moderne et réactive :

Frontend (Symfony + Nuxt) : Application frontend développée avec Symfony et Nuxt.js, offrant une interface utilisateur intuitive et réactive.

Cartographie Interactive : Intégration de services de cartographie pour afficher les points de vente sur des cartes interactives.

Les utilisateurs accèdent à l'application via une infrastructure sécurisée :

CloudFront CDN : Réseau de diffusion de contenu utilisé pour distribuer les contenus statiques et dynamiques à faible latence.

F5 WAF : Pare-feu applicatif web utilisé pour protéger l'application contre les menaces et les attaques.

Le flux de données dans le projet Tarkett est conçu pour garantir la fiabilité, la sécurité et la performance de l'application. En utilisant des technologies modernes et des pratiques de gestion efficaces, le projet vise à offrir une expérience utilisateur optimale tout en assurant la sécurité et la maintenabilité des données.

1.4.1. Etude de l'existant

1.4.1.1. Migration du cœur Symfony de v5.3 vers v6.4

Le site Tarkett Locator a été sur la version 5.3 de Symfony, ce qui a posé le problème que cette version n'était plus supportée soit au niveau de fixation des bugs, et aussi les bugs de sécurité, Ce qui nécessitait une migration vers la version Symfony 6.4 pour bénéficier à la fois des évolutions apportées avec cette version et de sa maintenabilité.

A mon intégration, une partie de la migration avait été déjà achevée dont :

- Une étude qui a été faite pour savoir ce qu'il sera modifié ou affecté par cette migration en se basant sur la documentation Symfony.
- Les modules à changer (les arborescences des fichiers).
- Comparaison entre la version 5.3 et 6.4 pour savoir les différences entre la structure de répertoires.
- Définir les actions à faire
- Migration partie Entity

- Modification des namespaces
- Mettre à jour les dépendances (Installer Flex de Symfony)
- Migration d'API Platform 2.6 vers 3.2
- Migration de PHP 7.3 vers 8.3

1.4.2. Solution Proposée par le projet

1.4.2.1. Migration du cœur Symfony de v5.3 vers v6.4 et d'API Platform v2.6 vers v3.2

La migration du cœur de Symfony de la version 5.3 à la version 6.4 et d'API Platform de la version 2.6 à la version 3.2 est une tâche complexe mais essentielle pour améliorer la sécurité, les performances et la maintenabilité de l'application Tarkett Locator.

1.5. Planification et Conduite du projet

La gestion de projet est une méthode conçue pour organiser un projet afin qu'il se déroule sans heurts du début à la fin. Lorsque la gestion de projet implique un ensemble de projets qui contribuent au même objectif, on parle de gestion de programme. En pratique, le projet évolue vers l'objectif final, qui doit s'adapter à des révisions fréquentes, mais être contrôlé et planifié. Par conséquent, toute modification doit rester planifiée. En particulier, les projets doivent rester dynamiques et équilibrer en permanence les contraintes de technologie, de coût et de temps.

Dans notre cas, il nous a été demandé d'assumer deux contraintes principales : la livraison d'un produit de qualité et la livraison dans les délais. Compte tenu de ces contraintes, de nos objectifs et de la nature de notre mission, nous avons choisi l'approche SCRUM avec les bonnes pratiques XP pour exécuter notre projet. La raison du choix de cette méthode est de diviser le projet en plusieurs sprints(itérations). A chaque

sprint, une partie du projet sera traitée et devra être livrée en fin du sprint.

1.5.1. Méthodologie SCRUM

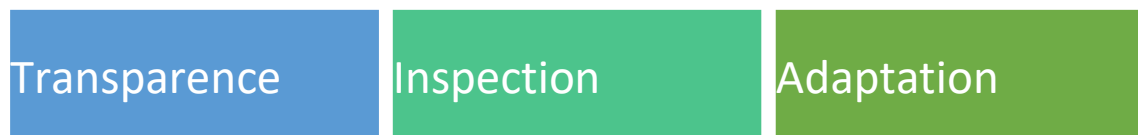
1.5.1.1. Description de la méthodologie SCRUM

Scrum (mêlée du Rugby) est une méthodologie dédiée à la gestion de projet avec pour principal objectif l'optimisation de la productivité et qui est apparue pour la première fois en 1986 dans une publication de Hirotaka Takeuchi et Ikujiro Nonaka intitulée *The New Product Development Game*, qui s'appliquait alors au monde industriel.

Scrum est un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible.

Scrum permet de travailler en équipe pour faire de l'amélioration continue sur des livraisons itératives incrémentales de produits afin de satisfaire vos clients.

Scrum est fondé sur la théorie du contrôle empirique de processus et soutenu par 3 piliers fondamentaux :



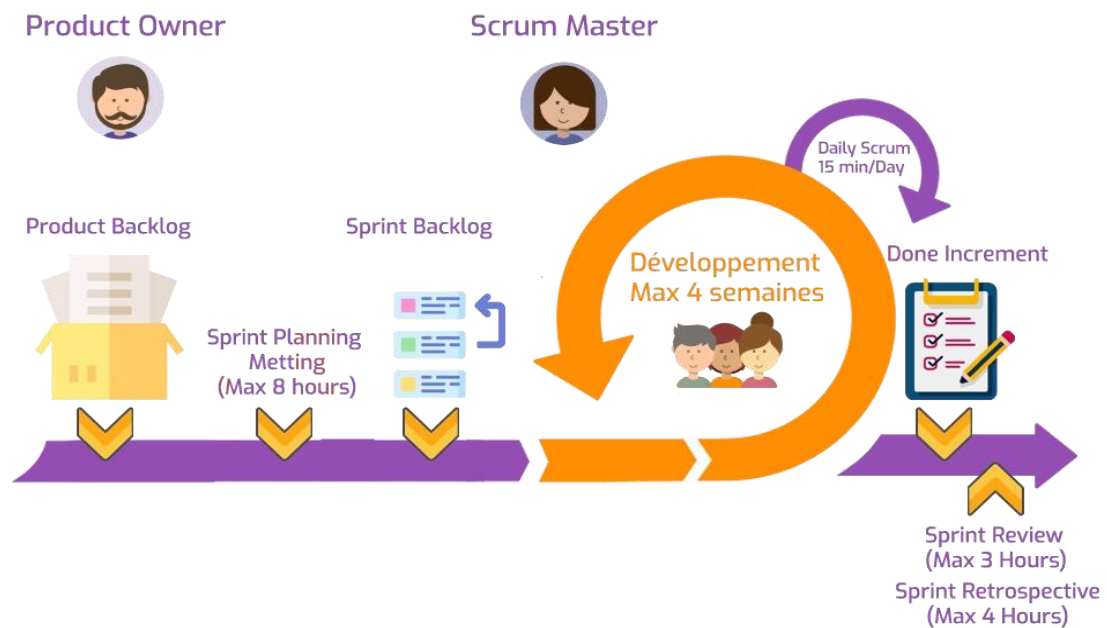


Figure 6: Processus Scrum

Répartition des rôles :

L'équipe Scrum est autoorganisée et multidisciplinaire, c'est-à-dire qu'elle choisit la meilleure façon d'effectuer le travail et elle possède toutes les compétences nécessaires pour mener à bien le projet. La flexibilité, la créativité et la productivité de l'équipe sont ainsi optimisées.

L'équipe Scrum se compose de :

Le Scrum Master est responsable de la compréhension, du respect et de la mise en œuvre de la méthode Scrum qu'il maîtrise parfaitement. Il veille au respect des principes et des valeurs de la méthodologie. C'est un facilitateur qui aide à améliorer la communication au sein des équipes et cherche à maximiser la productivité et l'expertise de l'équipe. Il est considéré comme l'entraîneur de l'équipe de développement.

Le Product Owner porte la vision du produit à réaliser. Il interagit avec l'équipe de développement qui doit suivre ses instructions. Il lui appartient de

hiérarchiser les fonctionnalités à développer ou à corriger, et de vérifier les fonctionnalités complétées. Il est responsable de la gestion du backlog produit.

L'équipe de développement est chargée de traduire les exigences définies par le propriétaire du produit en fonctionnalités utilisables. Il est multidisciplinaire et possède toutes les compétences nécessaires pour exécuter un projet sans avoir recours à des services externes. Ses membres comprennent des architectes, des développeurs, des testeurs, etc. La taille idéale d'une équipe de développement est de 3 à 9 personnes. Il n'y a pas de notion de hiérarchie, toutes les décisions sont prises ensemble.

Les divers événements :

Le cycle de vie d'un projet Scrum est cadencé par une série de réunions définies avec précision et limitées dans le temps.

Le product backlog : Il s'agit d'une liste hiérarchisée des exigences initiales du client concernant le produit à réaliser. Ce document évolue sans cesse durant le projet, en fonction des besoins du client. Le product owner est responsable du product backlog.

Le Sprint backlog : C'est le plan détaillé de la réalisation de l'objectif du Sprint, défini lors de la réunion de planification du Sprint. Le Sprint backlog est mis à jour régulièrement par l'équipe afin d'avoir une vision précise de la progression du Sprint.

Planification du sprint : Au cours de cette réunion, l'équipe de développement détermine les éléments prioritaires du Product Backlog (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du Sprint, en accord avec le Product Owner. Cette réunion permet donc à l'équipe d'établir les éléments qu'elle traitera au cours du Sprint et comment elle procédera.

Sprint : est une période courte et limitée dans le temps (2 semaines dans notre cas) où une équipe Scrum travaille pour compléter une quantité de travail définie. Les sprints sont au cœur même de Scrum et des méthodologies agiles, et obtenir les bons sprints aidera votre équipe agile à livrer de meilleurs logiciels avec moins de maux de tête.

Mêlée quotidienne : Cette réunion quotidienne de 15 minutes est très

importante. Elle se fait debout (d'où son nom anglais de "stand-up meeting") afin d'éviter de s'éterniser et de **permettre de rester agile**. Le but est de faire un point sur la progression journalière du Sprint. Elle permet à l'équipe de synchroniser ses activités et de faire un plan pour les prochaines 24 heures. La mêlée a lieu à la même heure et au même endroit chaque jour.

Revue de sprint : consiste à démontrer le travail acharné de toute l'équipe : concepteurs, développeurs et Product Owner. Les membres de l'équipe se rassemblent autour d'un bureau pour des démonstrations informelles et décrivent le travail qu'ils ont effectué pour cette itération. C'est le moment de poser des questions, d'essayer de nouvelles fonctionnalités et de donner votre avis. Le partage du succès est un élément important de la construction d'une équipe agile.

Rétrospective de sprint : Après la revue du Sprint, cette réunion est l'occasion de déterminer ce qui peut être amélioré suite au Sprint écoulé (productivité, qualité, efficacité, conditions de travail, etc.). Cette réunion repose sur le principe d'amélioration continue.

L'incrément : Il s'agit de l'ensemble des éléments terminés du product backlog pour le Sprint en cours, ainsi que ceux des Sprints précédents. L'incrément doit fonctionner et être utilisable.

Nous utilisons l'outil d'agilité en ligne :

T	Pr	État	Clé	Résumé	Rapporteur	Attribution	Mise À Jour	Demo Date	Étiquettes	Versions Corrigées	Point Effort
			AD-8694	[CORA][Dossier de chiffrage] Commande vide et ligne de produit sans IAM après mise à jour chiffrage SIDEVA	Florian Peuteuil	Berrahma Ouafae	07/juin/22 11:56 AM		BONUS, Bonus-S82, Bonus-S83, CORA		
			AD-9033	BUG- [Evolution Dossier][Cora] Suppression de la fonctionnalité "Annuler" unitaire par OE avec IAM	TURBAN	TURBAN	jeudi 10:02 AM			3.114.0	
			AD-8938	implémentation appels multi références	Romain Villaumé	Romain Villaumé	mercredi 10:04 AM		BONUS-S80, BONUS-S81	3.114.0	1.0
			AD-9009	[ODIGO] Pas de Code Commercial + Login et pass mal codés	Laurent Coutaye	Laurent Coutaye	mardi 10:23 AM			3.114.0	
			AD-8997	[ODIGO] Package Odigo affecté à des comptes déjà existant	Laurent Coutaye	Laurent Coutaye	Aujourd'hui 9:36 AM			3.114.0	
			AD-9013	[Verriers] Gestion de la disponibilité Vs Available quantity	Bruno CHAUVEAU	Bruno CHAUVEAU	mercredi 9:55 AM			3.114.0	
			AD-9002	[Verriers] Commandes affichées en double avec double IAM	Bruno CHAUVEAU	Bruno CHAUVEAU	mercredi 10:15 AM			3.114.0	
			AD-8989	[Verriers] Bloquer l'affichage du bouton "+ de dispo"	Bruno CHAUVEAU	Bruno CHAUVEAU	mercredi 10:12 AM			3.114.0	
			AD-8928	[CORA][Punchout DMS Fiducial] Autoriser l'ajout au panier des pièces CORA indisponibles	Laurent Coutaye	Laurent Coutaye	mercredi 10:08 AM		CORA, Fiducial, RETARD-S82	3.114.0	
			AD-7247	Envoi des mails récap	Karim Rehahla	Karim Rehahla	jeudi 9:57 AM		RETARD-S82	3.114.0	3.0

10 demandes Actualiser

Figure 7: Logo Jira

1.5.2. Les Outils de communication

Compte tenu de la répartition géographique des différents membres, l'utilisation d'outils de communication est essentielle au fonctionnement du travail d'équipe. Dans certains cas, l'un des membres est dans l'impossibilité de se rendre au bureau et il peut continuer à communiquer avec les autres membres et effectuer ses tâches à distance. Étant donné que l'entreprise est spécialisée dans le développement Web, il n'y a aucune restriction sur le travail à distance, et les réunions d'équipe et les présentations des progrès peuvent se faire via des outils de vidéoconférence ou de discussion actuels. Les deux majeurs outils utilisés sont :



Figure 8: Processus Microsoft Teams

1.5.3. Conclusion

Dans ce chapitre, j'ai défini le contexte général de mon projet en présentant, tout d'abord, l'entreprise d'accueil (SQLI) qui accompagne ses clients dans la mise en œuvre et le pilotage de leur transformation digitale. J'ai également introduit le contexte de mon projet basé sur le Framework Symfony. Finalement, une présentation de la conduite du projet qui est une combinaison de JIRA et du Scrum.

CHAPITRE 2 : ETUDE FONCTIONNELLE

2.1. Introduction

Ce chapitre est consacré à l'étude fonctionnelle et à la conception du système pour le projet de migration du composant Tarkett Locator. L'objectif est de comprendre les besoins fonctionnels de l'application, de définir les spécifications nécessaires et de concevoir une architecture optimale pour répondre à ces exigences.

2.2 Vérification des dépendances obsolètes et des vulnérabilités de sécurité (Symfony 5.3)

2.2.1 Introduction

Avant d'entamer la migration de Symfony 5.3 à 6.4, il est crucial de vérifier les dépendances obsolètes et les vulnérabilités de sécurité. Cette étape permet de garantir que l'application est sécurisée et que toutes les dépendances utilisées sont compatibles avec la nouvelle version de Symfony.

2.2.2 Utilisation de composer audit

Composer offre une commande intégrée pour vérifier les vulnérabilités de sécurité dans les dépendances du projet :

php composer.phar audit

```
root@56688e589700:/var/www/tarkett-locator# php composer.phar audit
```

```
Found 6 security vulnerability advisories affecting 6 packages:
```

Package	api-platform/core
Severity	high
CVE	CVE-2023-25575
Title	CVE-2023-25575: Secured properties may be accessible within collections
URL	https://github.com/api-platform/core/security/advisories/GHSA-vr2x-7687-h6qv
Affected versions	>=2.6.0,<2.7.10 >=3.0.0,<3.0.12 >=3.1.0,<3.1.3
Reported at	2023-02-28T10:37:00+00:00
Package	guzzlehttp/psr7
Severity	medium
CVE	CVE-2023-29197
Title	Improper header validation
URL	https://github.com/guzzle/psr7/security/advisories/GHSA-wxmh-65f7-jcvw
Affected versions	>=2,<2.4.5 <1.9.1
Reported at	2023-04-17T16:00:00+00:00
Package	symfony/http-kernel
Severity	medium
CVE	CVE-2022-24894
Title	CVE-2022-24894: Prevent storing cookie headers in HttpCache
URL	https://symfony.com/cve-2022-24894
Affected versions	>=2.0.0,<2.1.0 >=2.1.0,<2.2.0 >=2.2.0,<2.3.0 >=2.3.0,<2.4.0 >=2.4.0,<2.5.0 >=2.5.0,<2.6.0 >=2.6.0,<2.7.0 >=2.7.0,<2.8.0 >=2.8.0,<3.0.0 >=3.0.0,<3.1.0 >=3.1.0,<3.2.0 >=3.2.0,<3.3.0 >=3.3.0,<3.4.0 >=3.4.0,<4.0.0 >=4.0.0,<4.1.0 >=4.1.0,<4.2.0 >=4.2.0,<4.3.0 >=4.3.0,<4.4.0 >=4.4.0,<4.4.50 >=5.0.0,<5.1.0 >=5.1.0,<5.2.0 >=5.2.0,<5.3.0 >=5.3.0,<5.4.0 >=5.4.0,<5.4.20 >=6.0.0,<6.0.20 >=6.1.0,<6.1.12 >=6.2.0,<6.2.6
Reported at	2023-02-01T08:00:00+00:00

+-----+		
Package	symfony/security-bundle	
Severity	medium	
CVE	CVE-2022-24895	
Title	CVE-2022-24895: Possible CSRF token fixation	
URL	https://symfony.com/cve-2022-24895	
Affected versions	>=2.0.0,<2.1.0 >=2.1.0,<2.2.0 >=2.2.0,<2.3.0 >=2.3.0,<2.4.0 >=2.4.0,<2.5.0 >=2.5	
	.0,<2.6.0 >=2.6.0,<2.7.0 >=2.7.0,<2.8.0 >=2.8.0,<3.0.0 >=3.0.0,<3.1.0 >=3.1.0,<3	
	.2.0 >=3.2.0,<3.3.0 >=3.3.0,<3.4.0 >=3.4.0,<4.0.0 >=4.0.0,<4.1.0 >=4.1.0,<4.2.0	
	>=4.2.0,<4.3.0 >=4.3.0,<4.4.0 >=4.4.0,<4.4.50 >=5.0.0,<5.1.0 >=5.1.0,<5.2.0 >=5.	
	2.0,<5.3.0 >=5.3.0,<5.4.0 >=5.4.0,<5.4.20 >=6.0.0,<6.0.20 >=6.1.0,<6.1.12 >=6.2.	
	0,<6.2.6	
Reported at	2023-02-01T08:00:00+00:00	
+-----+		
+-----+		
Package	symfony/twig-bridge	
Severity	medium	
CVE	CVE-2023-46734	
Title	CVE-2023-46734: Potential XSS vulnerabilities in CodeExtension filters	
URL	https://symfony.com/cve-2023-46734	
Affected versions	>=2.0.0,<2.1.0 >=2.1.0,<2.2.0 >=2.2.0,<2.3.0 >=2.3.0,<2.4.0 >=2.4.0,<2.5.0 >=2.5	
	.0,<2.6.0 >=2.6.0,<2.7.0 >=2.7.0,<2.8.0 >=2.8.0,<3.0.0 >=3.0.0,<3.1.0 >=3.1.0,<3	
	.2.0 >=3.2.0,<3.3.0 >=3.3.0,<3.4.0 >=3.4.0,<4.0.0 >=4.0.0,<4.1.0 >=4.1.0,<4.2.0	
	>=4.2.0,<4.3.0 >=4.3.0,<4.4.0 >=4.4.0,<4.4.51 >=5.0.0,<5.1.0 >=5.1.0,<5.2.0 >=5.	
	2.0,<5.3.0 >=5.3.0,<5.4.0 >=5.4.0,<5.4.31 >=6.0.0,<6.1.0 >=6.1.0,<6.2.0 >=6.2.0,	
	<6.3.0 >=6.3.0,<6.3.8	
Reported at	2023-11-10T08:00:00+00:00	
+-----+		

Package	twig/twig	
Severity	high	
CVE	CVE-2022-39261	
Title	Possibility to load a template outside a configured directory when using the	
	filesystem loader	
URL	https://symfony.com/blog/twig-security-release-possibility-to-load-a-template-outside-a-configured-directory-when-using-the-file-system-loader	
Affected versions	>=1.0.0,<1.44.7 >=2.0.0,<2.15.3 >=3.0.0,<3.4.3	
Reported at	2022-09-28T10:36:08+00:00	

Found 1 abandoned package:		

Abandoned Package	Suggested Replacement	

phpunit/php-token-stream	none	

Figure 9: Les vulnérabilités de sécurité et les dépendances abandonné

Nous constatons ici que 6 vulnérabilités de sécurité proviennent des dépendances Symfony (v5.3), il est donc nécessaire de mettre à jour Symfony de la version 5.3 à la version 5.4.

2.3 la migration du Symfony 5.3 vers 5.4:

Changement les versions des dépendances du Symfony en remplaçant '5.3' avec '5.4' dans le fichier **composer.json** qui est responsable de spécification des versions du projet.

Exemple:

```
- "symfony/xxxx": "5.3.*",
+ "symfony/xxxx": "5.4.*",
```

Exécuter la mise à niveau:

php composer.phar update symfony/* --with-all-dependencies

Résolution des problèmes de sécurité :

```

root@b4754c04617f:/var/www/tarkett-locator# php composer.phar audit
Found 3 security vulnerability advisories affecting 3 packages:
+-----+
| Package      | api-platform/core | | |
| Severity     | high              |
| CVE          | CVE-2023-25575    |
| Title        | CVE-2023-25575: Secured properties may be accessible within collections |
| URL          | https://github.com/api-platform/core/security/advisories/GHSA-vr2x-7687-h6qv |
| Affected versions | >=2.6.0,<2.7.10|>=3.0.0,<3.0.12|>=3.1.0,<3.1.3 |
| Reported at   | 2023-02-28T10:37:00+00:00 |
+-----+
+-----+
| Package      | guzzlehttp/psr7    | |
| Severity     | medium             |
| CVE          | CVE-2023-29197    |
| Title        | Improper header validation |
| URL          | https://github.com/guzzle/psr7/security/advisories/GHSA-wxmh-65f7-jcvw |
| Affected versions | >=2,<2.4.5|<1.9.1 |
| Reported at   | 2023-04-17T16:00:00+00:00 |
+-----+
+-----+
| Package      | twig/twig          | | |
| Severity     | high               |
| CVE          | CVE-2022-39261    |
| Title        | Possibility to load a template outside a configured directory when using the |
|              | filesystem loader  |
| URL          | https://symfony.com/blog/twig-security-release-possibility-to-load-a-template-outside-a-configured-directory-when-using-the-file-system-loader |
| Affected versions | >=1.0.0,<1.44.7|>=2.0.0,<2.15.3|>=3.0.0,<3.4.3 |
| Reported at   | 2022-09-28T10:36:08+00:00 |
+-----+

```

Figure 10: Les vulnérabilités de sécurité

Tous les avertissements indiquaient que les versions de dépendances mentionnées étaient affectées. La solution est de mettre à niveau les versions vers la version la plus élevée compatible avec notre projet à l'heure actuelle, et qui ne contient pas de faille de sécurité.

API-Platform/core:

Secured properties may be accessible within collections

High dunglas published GHSA-vr2x-7687-h6qv on Feb 28, 2023

Package	Affected versions	Patched versions
php-api-platform/core (Composer)	< 2.7.10	2.7.10
	< 3.0.12	3.0.12
	< 3.1.3	3.1.3
	>= 2.6.0	2.7.10

Severity
High 7.7 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	None
Availability	None

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

CVE ID
CVE-2023-25575

Weaknesses
CWE-842

Credits

Toflar	Analyst
soyuka	Analyst

Description

Impact

Resource properties secured with the `security` option of the `ApiPlatform\Metadata\ApiProperty` attribute can be disclosed to unauthorized users. The problem affects most serialization formats, including raw JSON, which is enabled by default when installing API Platform. Custom serialization formats may also be impacted. Only collection endpoints are affected by the issue, item endpoints are not. The JSON-LD format is not affected by the issue.

The result of the security rule is only executed for the first item of the collection. The result of the rule is then cached and reused for the next items. This bug can leak data to unauthorized users when the rule depends on the value of a property of the item. This bug can also hide properties that should be displayed to authorized users.

Patches

This issue impacts the 2.7, 3.0 and 3.1 branches. Upgrade to v2.7.10, v3.0.12 or v3.1.3.

Workarounds

Replace the `cache_key` of the context array of the Serializer inside a custom normalizer that works on objects if the security option of the `ApiPlatform\Metadata\ApiProperty` attribute is used.

References

No.

Figure 11: Détails de vulnérabilité de sécurité d'API Platform

La solution :

Mise à niveau du package vers la version la plus élevée compatible avec notre projet maintenant et ne contenant pas de vulnérabilité de sécurité.

Nous rechercherons dans le référentiel de dépendances [packagist](https://packagist.org).

On va dans la barre de droite pour voir toutes les versions, on choisit notre version actuelle (2.6.5).

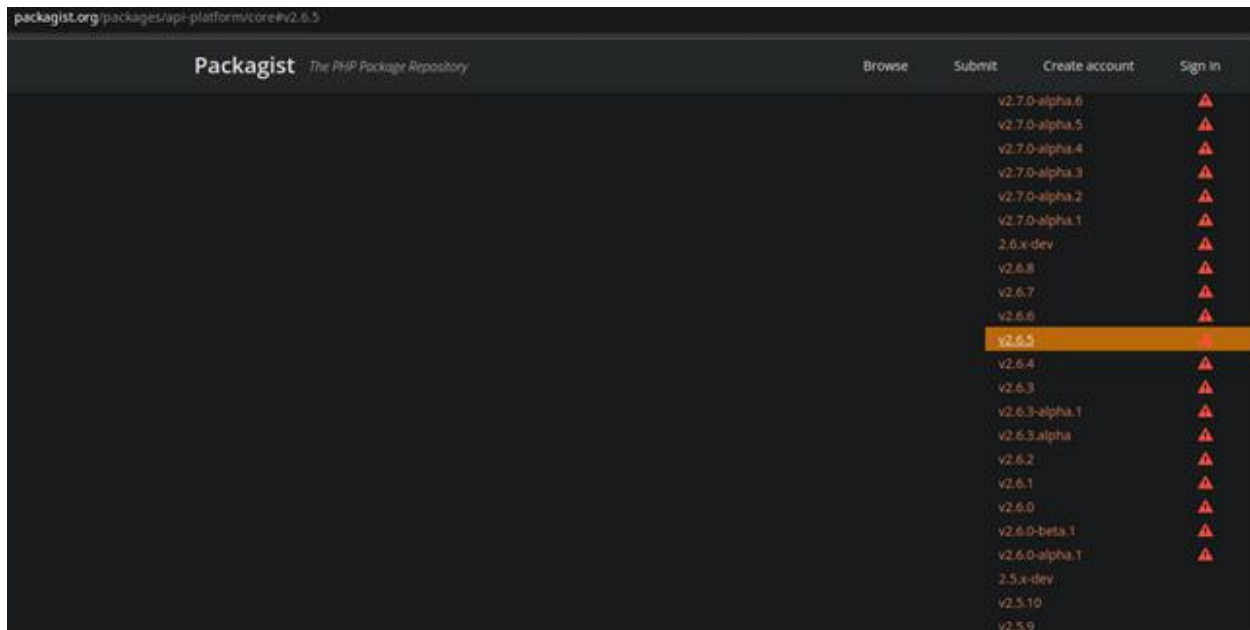


Figure 12: API Platform packagist.org

Nous choisirons version par version par ordre croissant jusqu'à la plus grande version compatible avec nos dépendances des projets.

Dans notre cas, nous passons à la version 2.7.18.

L'étape pour changer la version des dépendances, il suffit de lance la commande du composer pour exécute la mise à niveau :

`php composer.phar require api-platform/core:2.7.18`

Cela étant, on fait une mise à niveau des bibliothèques restantes si ce le cas.

`php composer.phar update`

Symfony/security-bundle

`#config/packages/security.yaml`

Le 'Guard' dans Symfony <=5.3 a été supprimée.

```
#
    guard:
#
    authenticators:
#
        - lexik_jwt_authentication.jwt_token_authenticator
```

Figure 13: Le fichier de configuration security.yaml

2.4 la migration du PHP 7.4 vers 8.1:

Introduction

La mise à jour de PHP de la version 7.4 à la version 8.0 est une étape essentielle pour bénéficier des améliorations de performance, de sécurité et des nouvelles fonctionnalités offertes par PHP 8.0. Cette section décrit les étapes suivies pour effectuer cette migration.

Tant que nous sommes dans un environnement **Docker**, nous devons apporter une modification dans Dockerfile :

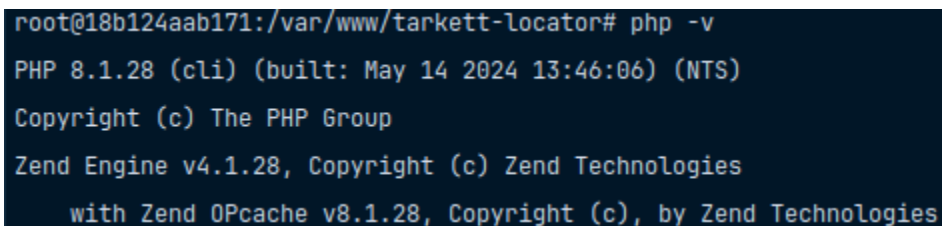
#Dockerfile:

#FROM php:7.4-fpm

FROM php:8.1-fpm

Puis on lance la recréation d'image du docker avec la commande :

docker-compose up -d --build --force-recreate



```
root@18b124aab171:/var/www/tarkett-locator# php -v
PHP 8.1.28 (cli) (built: May 14 2024 13:46:06) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.28, Copyright (c) Zend Technologies
with Zend OPcache v8.1.28, Copyright (c), by Zend Technologies
```

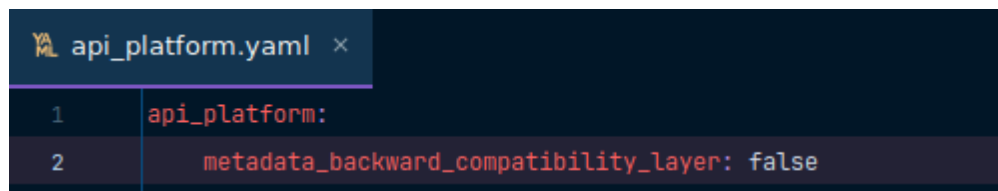
Figure 14: La version du PHP

Et ici nous obtenons la version PHP 8.1 avec succès.

La compatibilité d'API Platform avec PHP 8:

Si nous voulons qu'API Platform prenne en charge le nouveau système de métadonnées, nous devons définir, '*metadata_backward_compatibility_layer*' en '*false*':

#config/packages/api_platform.yaml:



```
api_platform:
  metadata_backward_compatibility_layer: false
```

Figure 15: Le fichier de configuration api_platform.yaml

Maintenant, nous devrions commencer à supprimer les annotations sur les attributs.

On prend exemple l'entité City.

L'entité : la classe dont lié à une table de la base de données.

Avant:

```
/**
 * @ApiResource(
 *     collectionOperations={
 *         "get",
 *         "post"={"security" = "is_granted('IS_AUTHENTICATED_FULLY')"}
 *     },
 *     itemOperations={
 *         "get",
 *         "put"={"security" = "is_granted('IS_AUTHENTICATED_FULLY')"},
 *         "patch"={"security" = "is_granted('IS_AUTHENTICATED_FULLY')"},
 *         "delete"={"security" = "is_granted('IS_AUTHENTICATED_FULLY')"}
 *     },
 * )
 * @ORM\Entity(repositoryClass=CityRepository::class)
 * @ApiFilter(SearchFilter::class, properties={"name"="ipartial", "stateProvince"="exact"})
 * @ApiFilter(OrderFilter::class, properties={"id", "name", "stateProvince.name", "default_direction" : "ASC"})
 */
class City
```

Figure 16: l'ancien Api resource

```
/**
 * @var integer
 *
 * @ORM\Id
 * @ORM\GeneratedValue
 * @ORM\Column(type="integer")
 *
 * @Groups({"city:read"})
 */
public int $id;
```

Figure 16: l'ancien annotation

Après:

```
#[ApiResource(  
    operations: [  
        new Get(),  
        new Post(security: 'is_granted("IS_AUTHENTICATED_FULLY")'),  
        new GetCollection(),  
        new Put(security: 'is_granted("IS_AUTHENTICATED_FULLY")'),  
        new Patch(security: 'is_granted("IS_AUTHENTICATED_FULLY")'),  
        new Delete(security: 'is_granted("IS_AUTHENTICATED_FULLY")'),  
    ]  
)]  
#[ORM\Entity(repositoryClass: CityRepository::class)]  
#[ApiFilter(SearchFilter::class, properties: ['name' => 'partial', 'stateProvince' => 'exact'])]  
#[ApiFilter(OrderFilter::class, properties: ['id', 'name', 'stateProvince.name', 'default_direction' => 'ASC'])]  
class City
```

Figure 17: le nouveau Api resource

```
#[ORM\Id]  
#[ORM\GeneratedValue]  
#[ORM\Column]  
#[Groups(['city:read'])]  
public int $id;
```

Figure 18: le nouveau attribut

Changement les espaces de nom (namespaces) :

Avant:

```
#use ApiPlatform\Core\Annotation\ApiResource;
```

Après:

```
use ApiPlatform\Metadata\ApiResource;  
use ApiPlatform\Metadata>Delete;  
use ApiPlatform\Metadata\Get;  
use ApiPlatform\Metadata\GetCollection;  
use ApiPlatform\Metadata\Patch;  
use ApiPlatform\Metadata\Post;  
use ApiPlatform\Metadata\Put;
```

Figure 19: les espaces de nom

Doctrine

Introduction

Doctrine est un ORM (Object-Relational Mapping) utilisé pour mapper les objets PHP aux tables de la base de données. La migration vers les attributs de PHP 8.0 permet d'améliorer la lisibilité et la maintenabilité du code en remplaçant les annotations par des attributs.

#config/packages/doctrine.yaml

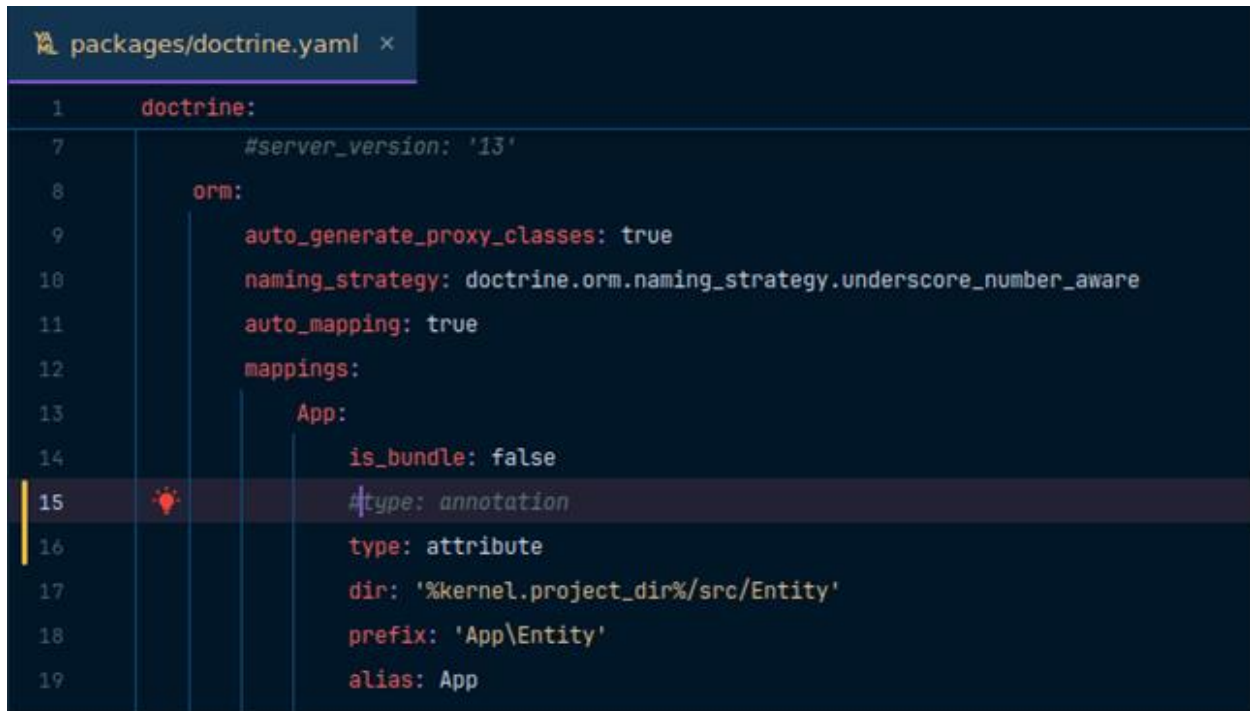


Figure 20: le fichier de configuration de Doctrine

La déclaration des variables:

Avant :

```
private string $name;

public function __construct(string $name)
{
    $this->name = $name;
}
```

Après:

```
public function __construct(private readonly string $name)
{
}
```

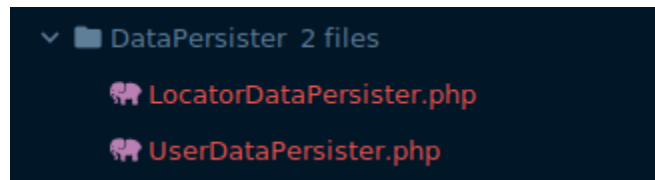
Figure 21: la déclaration d'un variable

Data persister :

Introduction

Dans API Platform, le Data Persister est utilisé pour gérer la persistance des données personnalisées. Avec la mise à jour vers API Platform 2.7 et ultérieur, le Data Persister a été remplacé par le Processor. Cette section décrit les étapes nécessaires pour effectuer cette migration.

Avant:



Après:

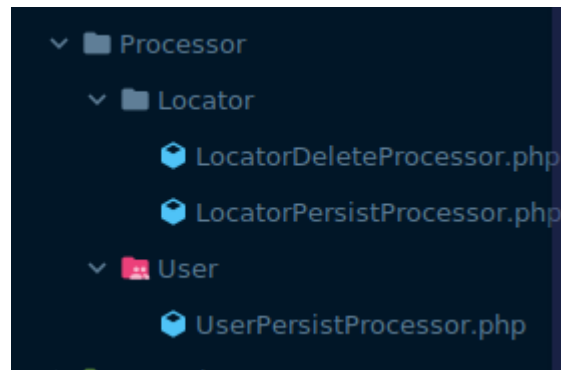


Figure 22: Data persister

Les résultats :

Aucun avis de vulnérabilité de sécurité trouvé.

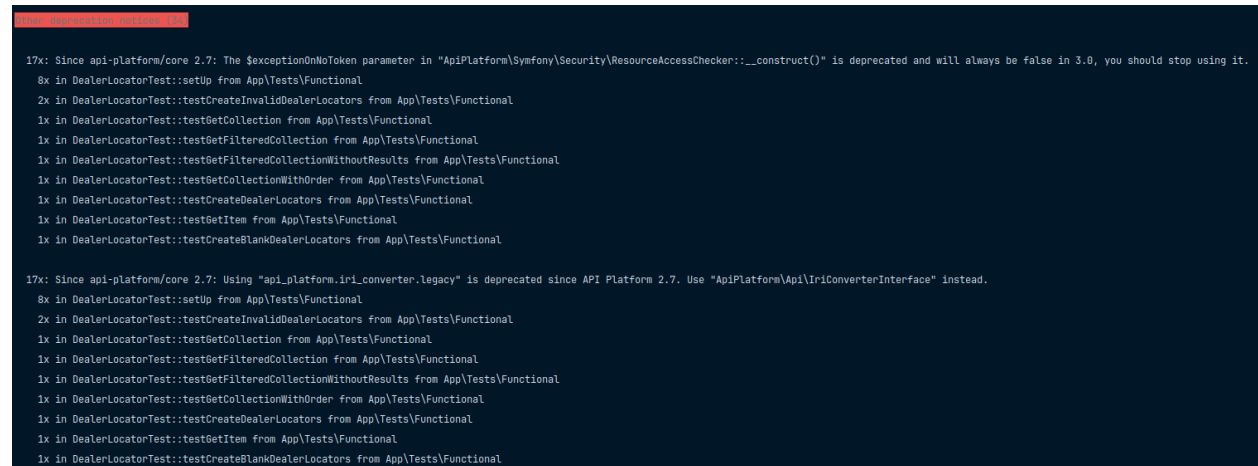
```
root@3ad4ba497dd0:/var/www/tarkett-locator# php composer.phar audit
No security vulnerability advisories found.
```

2.5 Vérification avec des Tests Unitaires

Après avoir effectué les migrations et les mises à jour des dépendances, il est crucial de vérifier que tout fonctionne correctement en exécutant des tests unitaires. Cela permet d'identifier et de corriger les éventuels problèmes introduits lors des modifications.

Exécuter la commande du test :

`php bin/phpunit tests/Functional`



```
17x: Since api-platform/core 2.7: The $exceptionOnNoToken parameter in "ApiPlatform\Symfony\Security\ResourceAccessChecker::__construct()" is deprecated and will always be false in 3.0, you should stop using it.
8x in DealerLocatorTest::setUp from App\Tests\Functional
2x in DealerLocatorTest::testCreateInvalidDealerLocators from App\Tests\Functional
1x in DealerLocatorTest::testGetCollection from App\Tests\Functional
1x in DealerLocatorTest::testGetFilteredCollection from App\Tests\Functional
1x in DealerLocatorTest::testGetFilteredCollectionWithoutResults from App\Tests\Functional
1x in DealerLocatorTest::testGetCollectionWithOrder from App\Tests\Functional
1x in DealerLocatorTest::testCreateDealerLocators from App\Tests\Functional
1x in DealerLocatorTest::testGetItem from App\Tests\Functional
1x in DealerLocatorTest::testCreateBlankDealerLocators from App\Tests\Functional

17x: Since api-platform/core 2.7: Using "api-platform.iri_converter.legacy" is deprecated since API Platform 2.7. Use "ApiPlatform\Api\IriConverterInterface" instead.
8x in DealerLocatorTest::setUp from App\Tests\Functional
2x in DealerLocatorTest::testCreateInvalidDealerLocators from App\Tests\Functional
1x in DealerLocatorTest::testGetCollection from App\Tests\Functional
1x in DealerLocatorTest::testGetFilteredCollection from App\Tests\Functional
1x in DealerLocatorTest::testGetFilteredCollectionWithoutResults from App\Tests\Functional
1x in DealerLocatorTest::testGetCollectionWithOrder from App\Tests\Functional
1x in DealerLocatorTest::testCreateDealerLocators from App\Tests\Functional
1x in DealerLocatorTest::testGetItem from App\Tests\Functional
1x in DealerLocatorTest::testCreateBlankDealerLocators from App\Tests\Functional
```

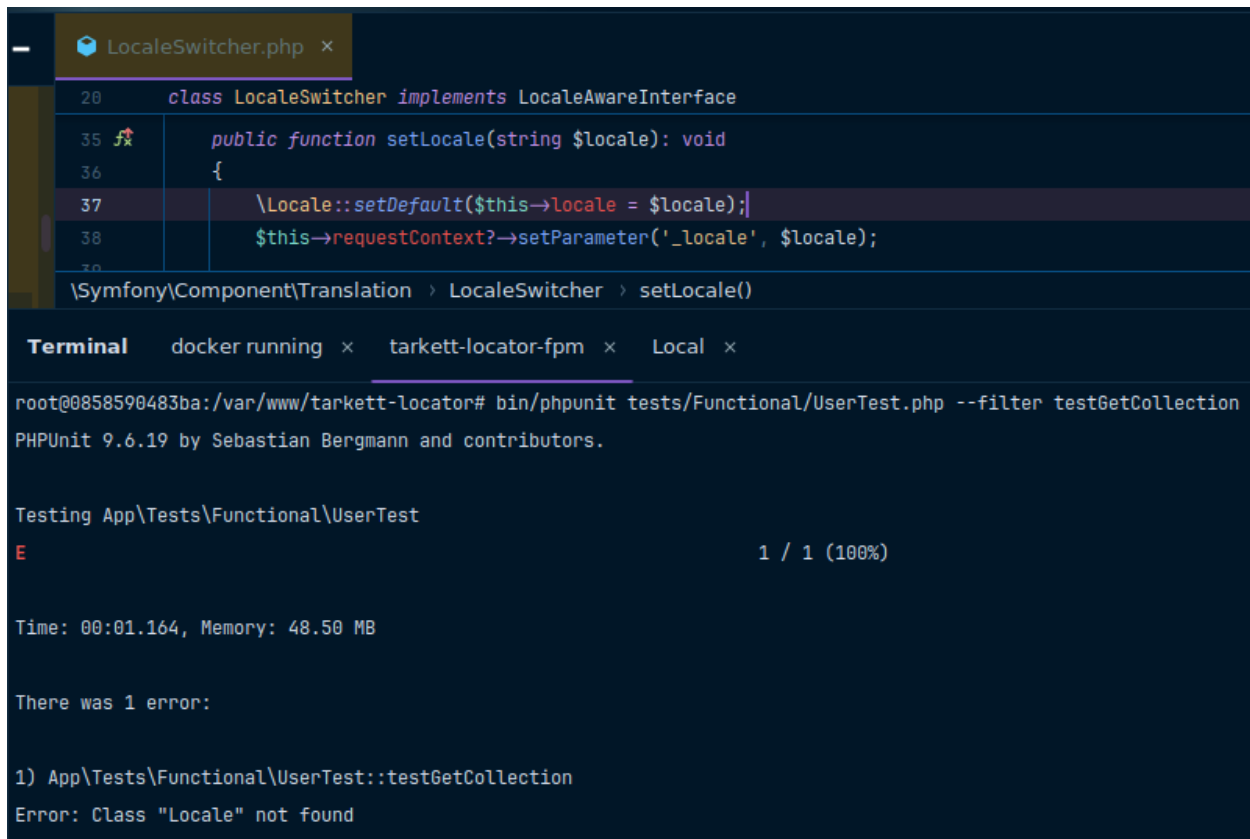
Figure 23: le résultat du test

Deux types de dépréciations persistent, mais nous n'avons pas accès pour les résoudre, avec plus de mises à niveau, nous le pouvons, parce que l'avertissement mentionne dépréciations dans le cœur du Symfony, et nous ne pouvons pas le modifier.

2.6 la migration du Symfony 5.4 vers 6.4:

Les premières étapes de la migration sont les mêmes mentionner dans la section '[la migration du Symfony 5.4 vers 6.4](#)'.

Exécuter les tests:



```
LocaleSwitcher.php x
20 class LocaleSwitcher implements LocaleAwareInterface
35 public function setLocale(string $locale): void
36 {
37     \Locale::setDefault($this->locale = $locale);
38     $this->requestContext?->setParameter('_locale', $locale);
39 }
\Symfony\Component\Translation > LocaleSwitcher > setLocale()

Terminal  docker running x  tarkett-locator-fpm x  Local x

root@0858590483ba:/var/www/tarkett-locator# bin/phpunit tests/Functional/UserTest.php --filter testGetCollection
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

Testing App\Tests\Functional\UserTest
E                                                                    1 / 1 (100%)

Time: 00:01.164, Memory: 48.50 MB

There was 1 error:

1) App\Tests\Functional\UserTest::testGetCollection
Error: Class "Locale" not found
```

Figure 24: conflit du version PHP

Dans ce cas, nous devons activer l'extension PHP *'intl'* dans la configuration PHP.

#Dockerfile

+ libicu-dev

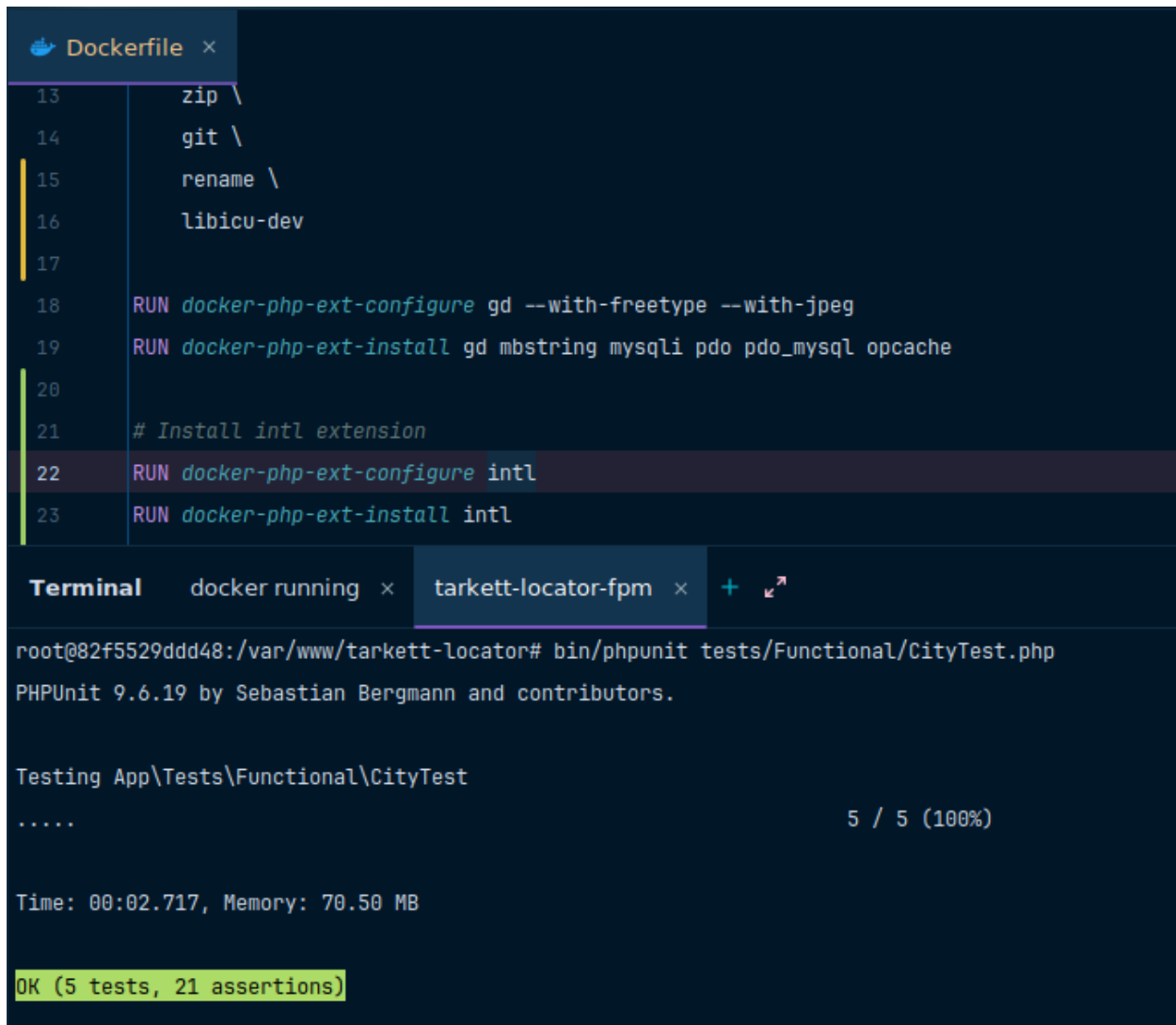
...

+ RUN docker-php-ext-configure intl

+ RUN docker-php-ext-install intl

Après avoir effectué les modifications, nous devons exécuter cette commande pour appliquer les modifications :

docker-compose up -d --build --force-recreate



The image shows a code editor with a Dockerfile and a terminal window. The Dockerfile contains the following lines:

```
13 zip \  
14 git \  
15 rename \  
16 libicu-dev  
17  
18 RUN docker-php-ext-configure gd --with-freetype --with-jpeg  
19 RUN docker-php-ext-install gd mbstring mysqli pdo pdo_mysql opcache  
20  
21 # Install intl extension  
22 RUN docker-php-ext-configure intl  
23 RUN docker-php-ext-install intl
```

The terminal window shows the output of running PHPUnit tests:

```
root@82f5529ddd48:/var/www/tarkett-locator# bin/phpunit tests/Functional/CityTest.php  
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.  
  
Testing App\Tests\Functional\CityTest  
.....  
5 / 5 (100%)  
  
Time: 00:02.717, Memory: 70.50 MB  
  
OK (5 tests, 21 assertions)
```

Figure 25: test OK

2.6.1 la migration d'API Platform du 2.7 vers 3.0:

Pour mettre à niveau API Platform vers la version 3.0, nous devons exécuter cette commande :

php composer.phar require api-platform/core:^3.0

Nous avons apporté ces modifications pour un problème de compatibilité :

Changement des espaces de noms :

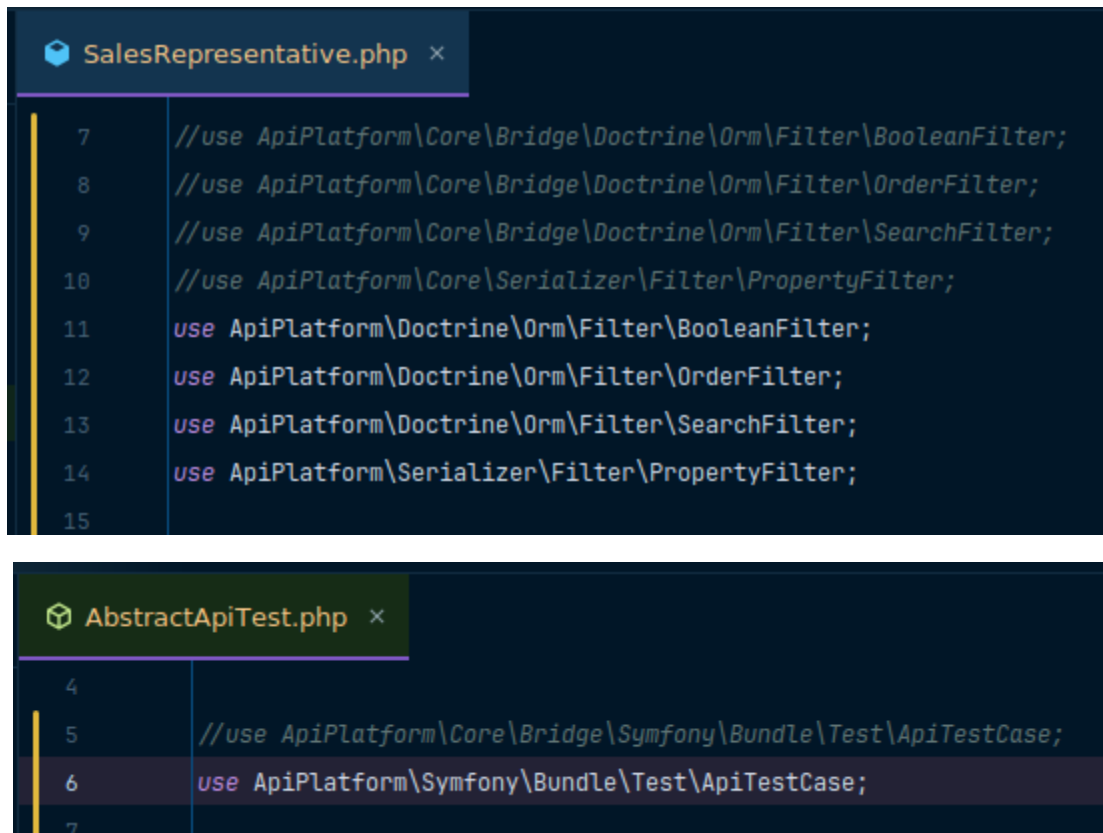


Figure 26: les espaces de nom

Enlever '*metadata_backward_compatibility_layer*':

#config/packages/api_platform.yaml

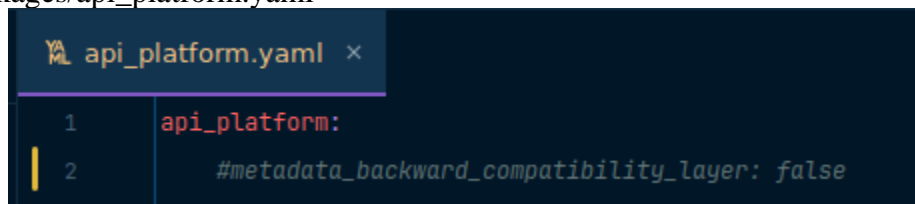


Figure 27 : le fichier de configuration d'API Platform

Les résultats :

Il reste deux dépréciations, nous expliquons déjà que nous ne pouvons pas les résoudre à ce stade :

```
root@82f5529ddd48:/var/www/tarkett-locator# bin/phpunit tests/Functional/UserTest.php
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

Testing App\Tests\Functional\UserTest
.....
6 / 6 (100%)

Time: 00:03.475, Memory: 66.50 MB

OK (6 tests, 18 assertions)

Remaining indirect deprecation notices (1)

1x: Method "ApiPlatform\State\ProcessorInterface::process()" might add "T2" as a native return type declaration in the future. Do the same in implementation "App\Processor\User\UserPersistProcessor" now to avoid errors or add an explicit @return annotation to suppress this message.
1x in UserTest::testCreateUser from App\Tests\Functional

Remaining indirect deprecation notices (1)

1x: The "Monolog\Logger" class is considered final. It may change without further notice as of its next major version. You should not extend it from "Symfony\Bridge\Monolog\Logger".
1x in UserTest::testGetCollection from App\Tests\Functional
```

Figure 28: les résultats du test

Nous pouvons désormais passer à API Platform 3.1.

2.6.2 la migration d'API Platform du 3.0 vers 3.2:

Pour mettre à niveau API Platform vers la version 3.2, nous devons exécuter cette commande :

```
php composer.phar require api-platform/core:^3.2
```

2.7 la migration du PHP 8.1 vers 8.3:

Introduction

La mise à jour de PHP de la version 8.1 à la version 8.3 est une étape essentielle pour bénéficier des améliorations de performance, de sécurité et des nouvelles fonctionnalités offertes par PHP 8.0. Cette section décrit les étapes suivies pour effectuer cette migration.

Tant que nous sommes dans un environnement **Docker**, nous devons apporter une modification dans Dockerfile :

#Dockerfile:

#FROM php:8.1-fpm

FROM php:8.3-fpm

Puis on lance la recréation d'image du docker avec la commande :

```
docker-compose up -d --build --force-recreate
```

```
root@f12a4d4dd863:/var/www/tarkett-locator# php -v
PHP 8.3.7 (cli) (built: Jun  6 2024 01:27:53) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.7, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.7, Copyright (c), by Zend Technologies
```

Figure 29: la version du PHP

2.7 Conclusion :

Ce chapitre avait pour but de présenter en détail les étapes et les procédures suivies pour migrer l'application Tarkett Locator de Symfony 5.3 à 6.4, d'API Platform 2.6 à 3.2, et de PHP 7.4 à 8.3. Nous avons abordé les vérifications des vulnérabilités, les mises à jour incrémentales, les adaptations du code, et les tests de compatibilité pour garantir la stabilité et la performance de l'application.

CHAPITRE 3 : TECHNOLOGIES ET OUTILS UTILISES

3.1. Introduction

Ce chapitre présente les différents Langages, Frameworks, Technologies et outils utilisées pour réaliser les tâches.

3.2. Technologies et Outils de développement

Ci-dessous l'ensemble des technologies et outils employés pour réaliser les différentes tâches :

3.2.1. Langages de programmation

3.2.1.1. PHP



Figure 30: Logo PHP

PHP (officiellement, ce sigle est un acronyme récursif pour PHP Hypertext Preprocessor) est un langage de programmation orienté objet, de scripts généralistes et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML, peut être exécuter en ligne de commande. PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, de nombreuses variantes Unix (incluant HP-UX, Solaris et OpenBSD), Microsoft Windows, macOS, RISC OS et d'autres encore. PHP supporte aussi la plupart des serveurs web actuels dont Apache, IIS et bien d'autres, la version utilisée dans le projet est : PHP 7.3.

3.2.2. Bibliothèques et Frameworks

3.2.2.1. *Symfony*



Figure 31: Logo Symfony

Symfony est un Framework qui représente un ensemble de composants (aussi appelés librairies) PHP autonomes qui peuvent être utilisés dans des projets web privé ou open source. Mais c'est également un puissant Framework PHP développé par une société française : SensioLabs. Il permet de réaliser des sites internet dynamiques de manière rapide, structurée, et avec un développement clair. Les développeurs peuvent travailler sur ce Framework très facilement, seuls ou en équipe, grâce à la facilité de prise en main.

3.2.2.3. *PHPUnit*



Figure 32: Logo PHPUnit

PHPUnit est un framework open source de tests unitaires dédié au langage de programmation PHP2. Il permet l'implémentation des tests de régression en vérifiant que les exécutions correspondent aux assertions prédéfinies.

3.2.3. Outils de développement

3.2.3.1. *Git*



Figure 33: Logo Git

Git est un système de contrôle de version distribué gratuit et open source conçue pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité. Git est facile à apprendre et a une petite empreinte avec des performances ultra-rapides. Il surclasse les outils SCM tels que Subversion, CVS, Perforce et ClearCase avec des fonctionnalités telles que la création de branches locales bon marché, des zones de mise en scène pratiques et plusieurs flux de travail.

3.2.3.2. *Github*



Figure 34: Logo Github

GitHub est une plateforme de développement collaboratif qui permet aux développeurs de gérer et de partager des projets de code source en utilisant le système de

contrôle de version Git. Elle offre des fonctionnalités pour le suivi des modifications, la collaboration en équipe, et la gestion des versions de projets logiciels.

3.2.3.3. Docker

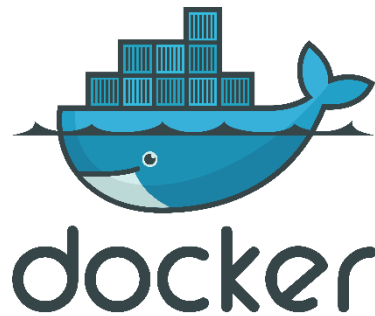


Figure 35: Logo Docker

Docker est une technologie de conteneurisation qui facilite la gestion de dépendance au sein d'un projet et ce, à tous les niveaux (développement et déploiement). Disponible sur Linux, Windows et Mac OS, le mécanisme de Docker se centre autour des conteneurs et de leur orchestration, et c'est en cela que la conteneurisation se différencie de la virtualisation.

3.2.3.4. Composer

Figure 36 : logo Composer



Composer est un outil de gestion des dépendances en PHP. Il vous permet de déclarer les librairies dont dépend votre projet et il les gère (installe/met à jour) pour vous.

3.2.3.5. PhpStorm



Figure 37: Logo PhpStorm

PhpStorm est un environnement de développement intégré (IDE) innovant, conçu par JetBrains pour les développeurs PHP et Web. Il prend en charge PHP toutes les versions du PHP, fournit une prévention des erreurs à la volée, la meilleure auto-complétions et la refactorisation du code, le débogage sans configuration et un éditeur HTML, CSS et JavaScript étendu... L'IDE fournit la saisie automatique intelligente du code, la coloration syntaxique, la configuration étendue du formatage du code, la vérification des erreurs à la volée, le pliage du code, la prise en charge des mélanges de langues, etc. La refactorisation automatisée traite votre code avec soin, contribuant à rendre les paramètres de projet globaux simples et sûrs.

3.2.4. Système de gestion de la base de données



Figure 38: Logo MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) open source qui utilise le langage SQL (Structured Query Language) pour interagir avec les données. Il est largement utilisé pour développer des applications web et gérer des bases de données, offrant fiabilité, performance et facilité d'utilisation.

3.3. Conclusion

Ce chapitre décrit un ensemble d'outils, que j'ai utilisé lors de l'implémentation des tâches. Cependant, le projet « Tarkett Locator » utilise d'autres Framework et des outils intéressants et puissants.

Le chapitre suivant présentera les tâches liées à la réalisation avec une étude détaillée.

CHAPITRE 4 : REALISATION DU PROJET

4.1. Introduction

Le présent chapitre est divisé en trois volets : le premier est destiné à la migration du Framework Symfony, le deuxième volet concerne la migration d'API Platform, et le troisième volet aborde la migration de PHP. Chaque volet comprend une description détaillée de la démarche suivie pour réaliser le travail.

4.2 Migration du Framework Symfony

Ce volet traite de la migration du cœur de Symfony de la version 5.3 à la version 6.4. La migration a été réalisée pour améliorer la sécurité, les performances et la maintenabilité de l'application Tarkett Locator.

Structure du projet Tarkett Locator au sein de Tarkett

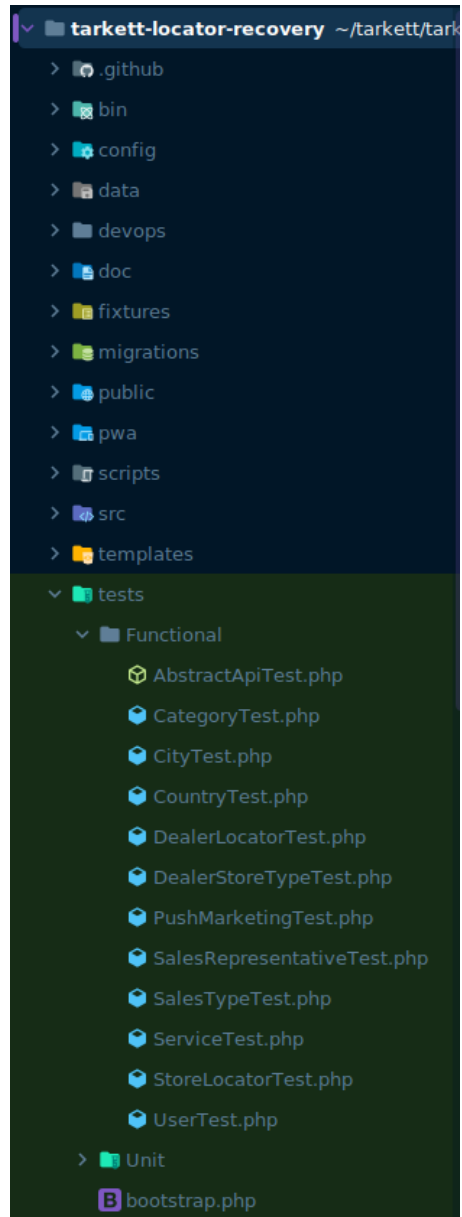


Figure 39 : l'architecture du composant Tarkett Locator

La version 6.4 de Symfony

```
root@f12a4d4dd863:/var/www/tarkett-locator# php composer.phar show symfony/framework-bundle
name      : symfony/framework-bundle
descrip.  : Provides a tight integration between Symfony components and the Symfony full-stack framework
keywords  :
versions  : * v6.4.9
released  : 2024-06-26, last week
type      : symfony-bundle
license   : MIT License (MIT) (OSI approved) https://spdx.org/licenses/MIT.html#licenseText
```

Figure 40: la version du Symfony

Exécution des tests

Pour exécuter les tests, nous avons deux manières :

Exécution niveau console « GitBash » dans notre cas :

```
root@f12a4d4dd863:/var/www/tarkett-locator# bin/phpunit tests/Functional/
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

Warning:      Abstract test case classes with "Test" suffix are deprecated (App\Tests\FUNCTIONAL\AbstractApiTest)

Testing /var/www/tarkett-locator/tests/Functional
..... 65 / 70 ( 92%)
..... 70 / 70 (100%)

Time: 00:31.364, Memory: 127.00 MB

OK (70 tests, 342 assertions)
```

Figure 41: les tests OK

Tous les tests sont OK.

Exécution niveau navigateur « Google Chrome » dans notre cas, voilà l'interface d'utilisateur marche comme il faut :

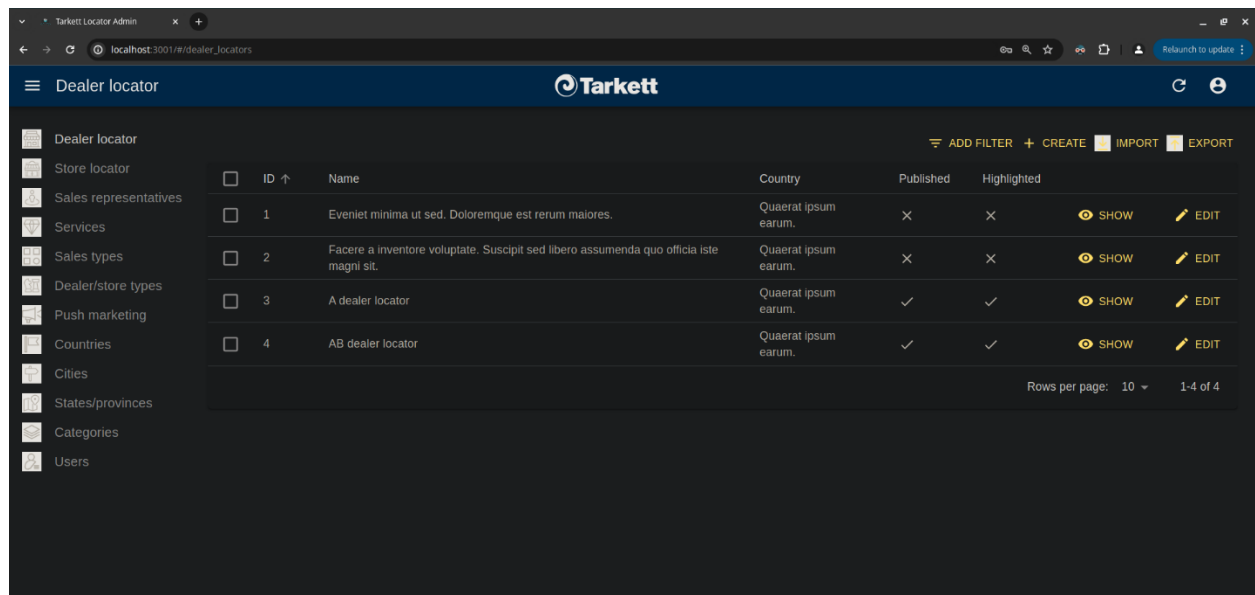


Figure 42: la page web du Tarkett Locator

4.3 Migration d'API Platform

La version 3.2 d'API Platform

```
root@f12a4d4dd863:/var/www/tarkett-locator# php composer.phar show api-platform/core
name       : api-platform/core
descrip.   : Build a fully-featured hypermedia or GraphQL API in minutes!
keywords   : Hydra, JSON-LD, api, graphql, hal, jsonapi, openapi, rest, swagger
versions   : * v3.2.0
released   : 2023-10-12, 8 months ago
type       : library
```

Figure 43: la version d'API Platform

L'application d'API Platform sur la page web "google chrome" :

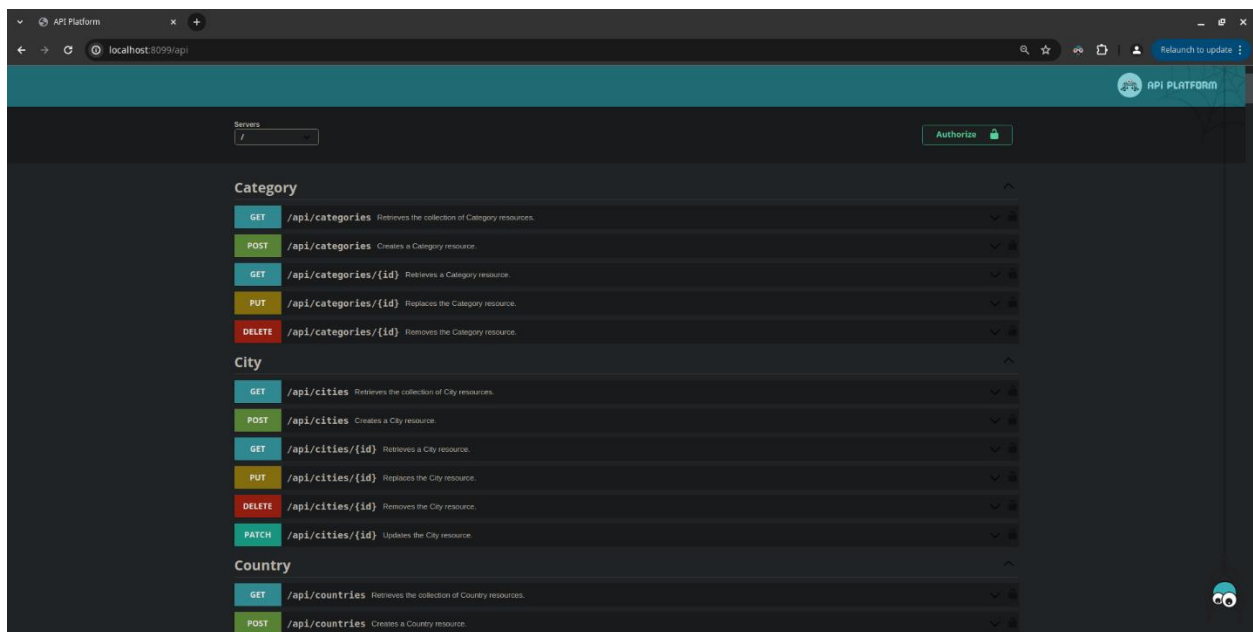


Figure 44: la page web du Swagger

4.4 Migration du PHP

Vérification de la version du PHP

```
root@f12a4d4dd863:/var/www/tarkett-locator# php -v
PHP 8.3.7 (cli) (built: Jun  6 2024 01:27:53) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.7, Copyright (c) Zend Technologies
with Zend OPcache v8.3.7, Copyright (c), by Zend Technologies
```

Figure 45: la version du PHP

4.4 Perspectives

Coordination avec l'Équipe DevOps :

En attendant, le retour de l'équipe DevOps, qui s'occupe de cette tâche, est crucial pour finaliser le déploiement.

Déploiement:

Le projet sera déployé sur le cloud AWS.

4.5. Conclusion

Les captures et les figures ont servi à mieux comprendre les interventions et le travail que j'ai réalisé durant ma période de stage.

CONCLUSION

Le présent rapport décrit les objectifs atteints durant mon stage de fin d'étude, portant sur la migration du composant Tarkett Locator. Ce projet visait à moderniser l'application en migrant le Framework Symfony de la version 5.3 à la version 6.4, en mettant à jour API Platform de la version 2.6 à la version 3.2, et en passant de PHP 7.3 à PHP 8.3.

Les principaux défis rencontrés incluaient la compréhension des nouvelles fonctionnalités et des différences entre les versions, ainsi que l'adaptation du code existant pour assurer la compatibilité. La première mission consistait à réaliser un audit des dépendances et à identifier les vulnérabilités de sécurité, ce qui a été crucial pour préparer la migration.

Ensuite, j'ai effectué une migration incrémentale, d'abord vers Symfony 5.4 puis vers 6.0, et enfin vers 6.4. Chaque étape a été suivie de tests de régression pour vérifier le bon fonctionnement de l'application. La mise à jour d'API Platform a suivi un processus similaire, en s'assurant que chaque version intermédiaire était correctement intégrée et testée.

La migration de PHP a impliqué la mise à jour des environnements de développement, de test et de production, ainsi que l'adaptation du code pour tirer parti des nouvelles fonctionnalités et des améliorations de performance offertes par PHP 8.3.

Les défis principaux étaient liés à la taille du projet et aux nombreuses dépendances externes. La gestion du temps et l'organisation méthodique ont été essentielles pour avancer efficacement. Grâce à la méthodologie SCRUM adoptée dans le projet, j'ai pu me concentrer sur les tâches prioritaires, avec le soutien constant de mes encadrants et de l'équipe.

Durant ce stage, j'ai acquis des compétences précieuses en travail d'équipe, en communication, en écriture de code propre et en respect des bonnes pratiques de

développement. J'ai également découvert la méthodologie agile SCRUM, qui s'est révélée très efficace pour gérer les projets complexes.

La migration du cœur de Symfony vers la version 6.4, d'API Platform vers la version 3.2, et de PHP vers la version 8.3 a été un succès, augmentant la performance et la sécurité de l'application. Pour l'avenir, nous visons à améliorer continuellement l'application et l'expérience utilisateur, à corriger les anomalies signalées par les utilisateurs, et à optimiser et stabiliser l'application.

Comme perspective, nous prévoyons de poursuivre les mises à jour et les optimisations, en visant à migrer éventuellement vers les versions encore plus récentes pour maintenir l'application à la pointe de la technologie.

WEBOGRAPHIE

- ❖ <https://symfony.com/doc>
- ❖ <https://api-platform.com/docs>
- ❖ <https://www.php.net/docs.php>
- ❖ <https://docs.docker.com>
- ❖ <https://docs.github.com>
- ❖ <https://dev.mysql.com/doc>