

Rapport 1^{ère} année d'apprentissage

Florian Guigue

Filière Informatique par apprentissage
Promotion 6
Année 2016-2017

Réalisation de missions autour du projet Mystique

Présentation des missions et évaluation des compétences



Maitre d'apprentissage : Nicolas Naline

Tuteur académique : Melchior Salgado

POLYTECH LYON
Domaine Scientifique de La Doua – 15, Boulevard Latarjet
69622 VILLEURBANNE CEDEX
Tél. (33) 04.72.43.12.24 - Fax. (33) 04.72.43.12.25
Web : <http://polytech.univ-lyon1.fr/>

Septembre 2017

Remerciements

Je tiens à remercier les différentes personnes qui m'ont suivi durant cette première année d'alternance.

Tout d'abord, j'adresse mes remerciements à mon maître d'apprentissage **Nicolas NALINE**, manager de l'équipe dans laquelle j'évolue, pour son aide et son suivi vis-à-vis de mon apprentissage et de ma formation.

Je remercie mon tuteur pédagogique **Melchior SALGADO** qui suit régulièrement l'avancer de mon apprentissage, aussi bien pendant les périodes de cours que d'entreprise.

Je tiens également à remercier les toutes les personnes de l'équipe dans laquelle je travaille pour m'avoir permis une bonne intégration et pour m'avoir fourni une aide et un support lors des difficultés que j'ai pu rencontrer.

Enfin, je tiens à remercier l'équipe pédagogique de Polytech, les professeurs et le personnel administratif qui me permettent de suivre la formation d'informatique en apprentissage.

Table des matières

Remerciements	2
Définitions	4
Introduction.....	5
1. Présentation	6
1.1. Présentation de l'entreprise	6
1.1.1. Atos	6
1.1.2. Worldline	6
1.2. Présentation de l'équipe	7
2. Activité de l'année.....	9
2.1. Première mission : Mise de place du packaging SQL.....	9
2.2. Deuxième mission : Variabilisation	10
2.3. Troisième mission : Dependency check	11
2.4. Quatrième mission : Batch-launcher	12
2.5. Cinquième mission : Refonte du site de Total	13
2.6. Sixième mission : Génération automatique de documentation	14
2.7. Septième mission : Extraction de données.....	15
3. Evolution des compétences	18
Conclusion.....	20
Références et sources	21
Résumé	22

Définitions

Tous les mots écrits en italique dans ce rapport se réfèrent à des définitions de cette page.

Annotations : Description d'un élément (classe, variable, méthode). Elle est toujours précédée d'un « @ » (Exemple : @Inject)

API : Interface de programmation composé d'un ensemble d'entités (classes, méthodes, fonction) offrant des services à d'autres logiciels

Backend : Partie du site effectuant les actions du côté du serveur

Batch : Script exécutant une succession de commandes

Frontend : Partie du site se situant du côté du client (par exemple la vue de la page web)

Plugin : Extension d'un logiciel apportant de nouvelles fonctionnalités

Product owner : Personne chargé de mettre en place et maintenir le backlog (liste des fonctionnalités du projet)

Wallet : Portefeuille électronique (Exemples : Paylib, PayPal)

Introduction

Dans le cadre de ma formation information en apprentissage à Polytech, j'ai effectué cette première année d'apprentissage au sein de l'entreprise Worldline. L'objectif de la formation en apprentissage est de permettre de suivre une formation informatique au sein de d'une école d'ingénieur tout en évoluant en parallèle dans le monde de l'entreprise.

L'objectif de ce rapport va être de montrer le travail effectué au cours de cette année d'apprentissage et ainsi montrer l'évolution de mes compétences dans le monde de l'entreprise.

Je commencerai par présenter mon entreprise d'accueil ainsi que l'équipe dans laquelle j'ai évolué au cours de cette année d'apprentissage. Je continuerai par une explication détaillée des différentes missions qui m'ont été donné, en précisant leur contexte et leurs enjeux. J'expliciterai ensuite l'évolution de mes compétences avant de terminer par mon retour d'expérience sur cette première année.

1. Présentation

1.1. Présentation de l'entreprise

1.1.1. Atos

Atos est la société mère de Worldline. C'est une Entreprise de Services du Numérique (ESN) réparti dans 72 pays à travers le monde avec 100 000 collaborateurs et affichant un chiffre d'affaire de 12 milliards d'euros en 2016. Atos se positionne à la première place européenne dans les domaines du Big Data, de la Cybersécurité, des supercalculateurs et de l'environnement de travail connecté. En plus de Worldline, Atos dispose d'autres filiales :

- **BULL** : regroupe les technologies d'Atos (matériels et logiciels)
- **CANOPY**: Cloud
- **BLUEKIWI** : Fournisseur de réseaux sociaux d'entreprise
- **ATOS CONSULTING** : Filiale d'Atos composée de consultants
- **ATOS WORLDGRIP** : Filiale d'Atos, fournit des projets d'intégration et des solutions de gestion intelligente de l'énergie en temps réel
- **UNIFY** : Marché des services informatiques

1.1.2. Worldline

Worldline a été d'abord fondé en 2004 en tant que division d'Atos pour la gestion des activités d'Atos concernant les terminaux de paiements et les paiements électroniques et en ligne. Ce n'est qu'à partir de 2013 que Worldline devient une entreprise en tant que filiale d'Atos. En 2016, la société compte plus de 8000 employés réparti dans 17 pays (figure 1) et totalise plus de 1,6 milliards d'euros de chiffre d'affaires.

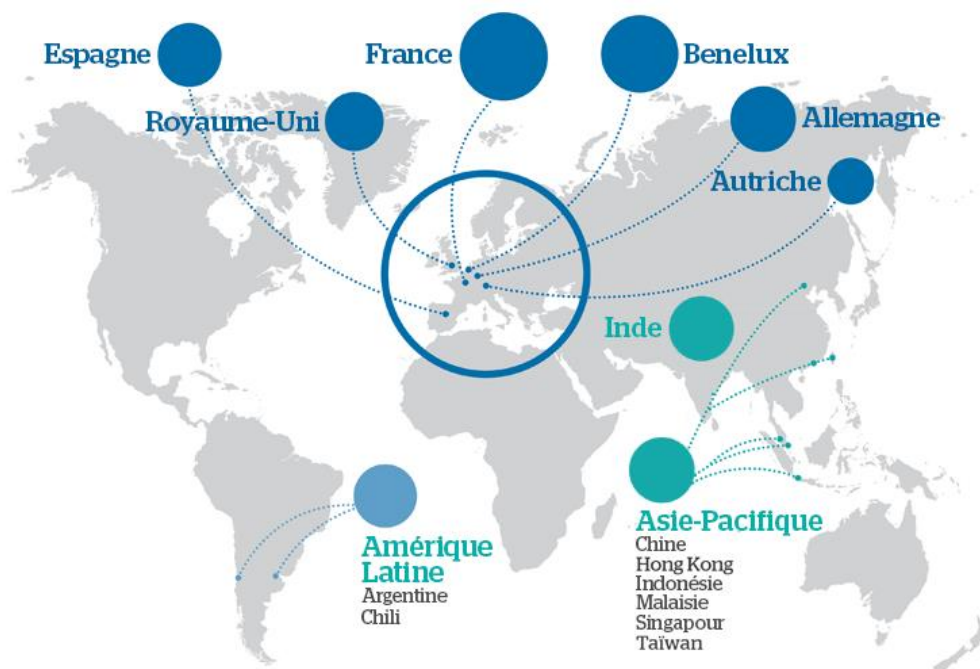


Figure 1 - Worldline dans le monde

Dirigé par Gilles Grapinet, Worldline est une société anonyme à conseil d'administration dont le siège social se situe à Bezons en France. Ce statut juridique se justifie par le fait que Worldline gère de grands projets avec des capitaux importants et cela leur permet une certaine souplesse au niveau de la gestion du capital.

Les activités de l'entreprise se regroupent autour de trois principaux produits et services (figure 2) :

- Services Commerçants & Terminaux de paiements qui représente 30% du chiffre d'affaire de Worldline,
- Traitement de transactions & logiciels de paiements qui représente 45% du chiffre d'affaire de Worldline,
- Mobilité & services web transactionnels qui représente 25% du chiffre d'affaire de Worldline.

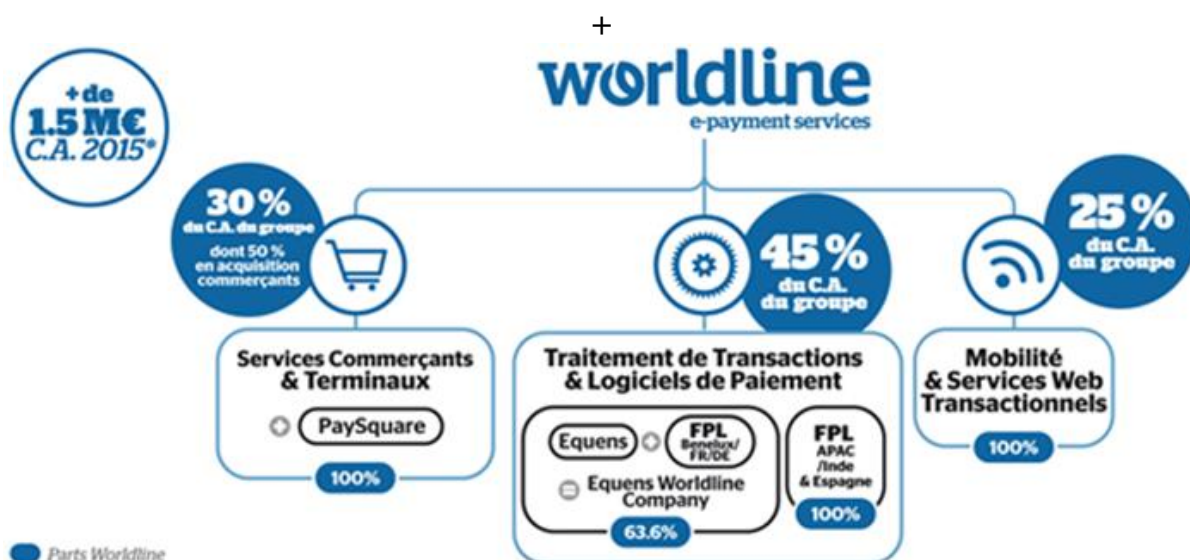


Figure 2 - Répartition du chiffre d'affaire en 2015

1.2. Présentation de l'équipe

L'équipe dans laquelle je travaille est managée par Nicolas NALINE et se situe dans le pôle Cloud Wallet (figure 3).

Lors de mon arrivé dans l'équipe, nous étions cinq :

- Nicolas NALINE en tant que manager,
- Antoine FOSSEY, Bruce PHILIPPE, Matthias LAO et moi-même en tant que développeurs.

Depuis un an, l'équipe a bien grandi puisqu'elle compte aujourd'hui neuf personnes dont un manager, un *product owner* et sept développeurs.

Dans cette équipe, un projet principal a été développé de A à Z : le projet Mystique. Mystique est un site web de gestion de *wallets*. Sa particularité est

d'être un site web générique, c'est-à-dire qu'une base commune du site sera utilisée puis adaptée aux besoins des différents clients pour chaque projet. Cela permet de fournir les fonctionnalités de Mystique sans avoir besoin de les redéveloppé pour chaque client.

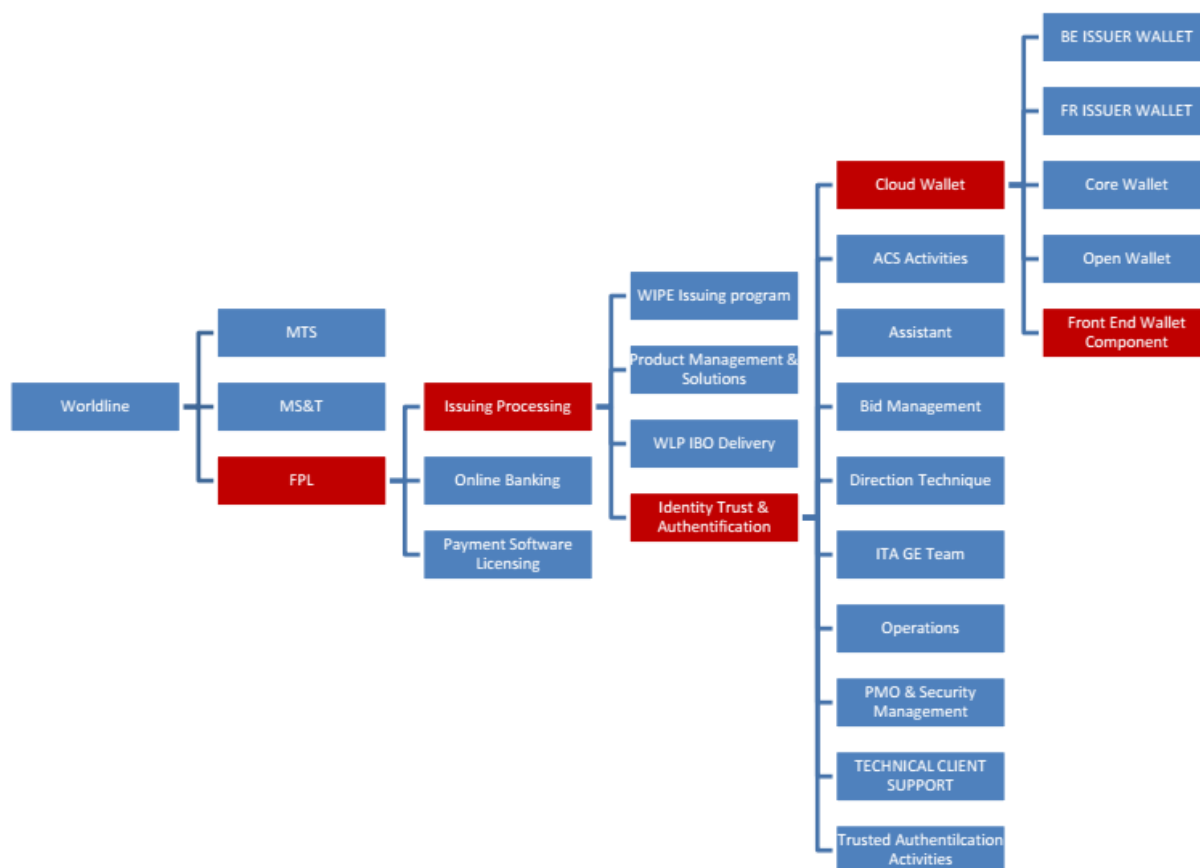


Figure 3 - Hiérarchie de l'équipe chez Worldline

Depuis début juillet 2017, une réorganisation du pôle a été mise en place. Toute l'équipe dont je fais partie change de projet et ne traitera plus d'aucun sujet lié à Mystique. Cependant, comme ce changement est très récent et que je n'ai effectué que peu de travaux sur les nouveaux sujets, je n'aborderai que les missions effectués avant cette réorganisation.

2. Activité de l'année

Le projet central sur lequel l'équipe que j'ai intégré travaille s'appelle le projet Mystique. Comme précisé plus haut, Mystique est un site web générique de gestion de *wallets*. Cela inclut donc que Mystique ne sera jamais livré en l'état et subira des modifications spécifiques demandé par les différents clients. Seul une base commune restera, c'est-à-dire les fonctionnalités techniques tels que la création de compte, l'ajout de cartes et d'adresses, la gestion des abonnements, etc...

Mystique a été développé autour de Maven. Cette précision est nécessaire car mes premières missions ont été basé sur l'utilisation de Maven.

Maven est un outil de gestion et d'automatisation de production des projets Java et Java EE. Grâce à cet outil, il est possible de gérer son projet de façon très poussé et de faciliter le déploiement et l'installation du projet. En plus des fonctionnalités déjà présentes avec l'outil, il est possible de rajouter des *plugins* et d'importer des dépendances pour son projet.

Au fil de cette partie, je vais expliquer les différentes missions qui m'ont été attribuées au cours de cette année.

2.1. Première mission : Mise de place du packaging SQL

Pour que Mystique puisse fonctionner, il est nécessaire d'avoir une base de données correctement construite. Pour construire cette base de données, des scripts SQL sont fournis afin de générer la structure et l'intégration des données dans la base. Seulement, lors de mon arrivée, il n'y avait que quelques gros scripts et mon but a été de découper ces gros scripts en plusieurs petits scripts. L'avantage de cette séparation en plusieurs scripts est de pouvoir différencier les commandes SQL qui permettront de construire la structure initiale de la base (tables, liens entre les tables, ...) de celles qui permettent de peupler la base.

J'ai divisé ces scripts en trois parties distinctes : les scripts ddl, les scripts dml et les scripts ext-dml.

Les scripts ddl sont la liste des commandes qui permettent de construire la structure de la base. C'est ici que l'on pourra trouver les commandes de création des différentes tables présentes dans la base, ainsi que les modifications apportées aux tables existantes.

Les scripts dml sont la liste des commandes qui permettent de remplir les tables avec les données. On pourra y retrouver les cartes bancaires, les adresses ou encore les *wallets* enregistrés sur le site.

Les scripts ext-dml sont des scripts similaires aux scripts dml. La différence est due au fait que ces scripts ne sont présents qu'à titre d'exemple et devront être modifiés par le client.

Une fois la séparation des scripts terminée, il fallait mettre en place un packaging de ces scripts sous forme de fichier compressé au format zip afin que ceux-ci puissent être livrés de manière pratique. Pour réaliser cela, j'ai utilisé le *plugin* « maven-assembly » qui permet de générer un fichier zip avec les fichiers désirés. Ainsi, à chaque installation du projet, Maven va s'occuper de générer un fichier compressé contenant l'arborescence de fichier ci-dessous (figure 4).

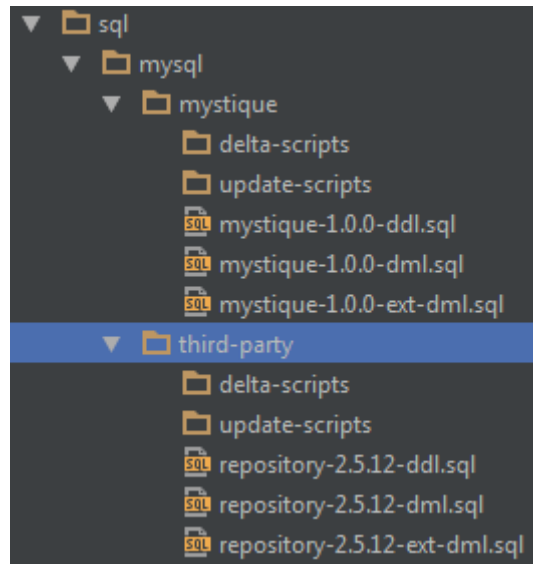


Figure 4 - Hiérarchie des scripts SQL

2.2. Deuxième mission : Variabilisation

Mystique pouvant être déployé sur différents environnements, certaines variables internes au projet tel que des URLs sont amenées à être modifiées. Le principe de cette mission de variabilisation est de faire en sorte de centraliser toutes les variables susceptibles d'être modifiées dans un même fichier afin que la modification des valeurs soit simplifiée.

Pour ce faire, j'ai créé deux dossiers : un dossier « template » pour centraliser les différents fichiers qui possèdent des valeurs à variabiliser et un dossier « filters » qui regroupe les valeurs de ces variables au sein de fichiers propriétés. Chaque fichier filtre correspond à un environnement de développement. Par exemple, dans la figure 5, le fichier « local-filter.properties » sera utilisé pour variabiliser les fichiers du dossier « template » lors d'un déploiement en local. De même pour les fichiers « docker-filter » et « acacia-filter » qui permettront de variabiliser sur les environnements docker et acacia. Le processus de variabilisation s'effectue avec l'aide de Maven. En spécifiant quel filtre utilisé lors de la compilation, Maven va s'occuper de remplacer les variables des fichiers de template par leur valeur inscrite dans le filtre correspondant.

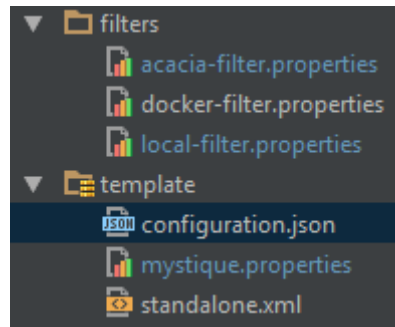


Figure 5 - Fichiers filtres et templates

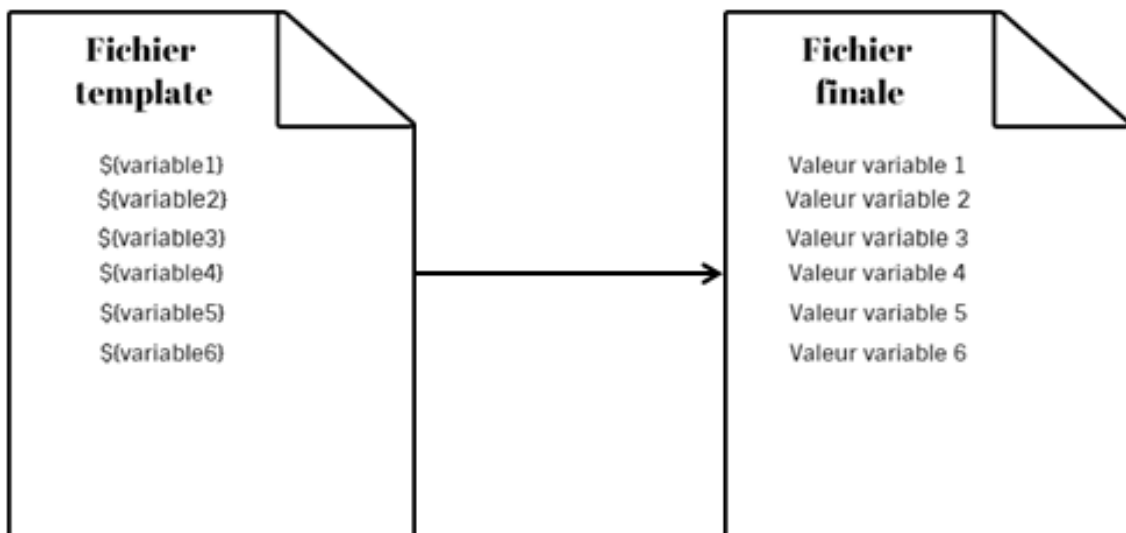


Figure 6 - Fonctionnement variabilisation

2.3. Troisième mission : Dependency check

Mystique utilise beaucoup de dépendances externes pour fonctionner. Cependant, ces dépendances peuvent avoir des vulnérabilités plus ou moins importantes. L'objectif de cette mission est donc de mettre en place un outil qui va scanner les différentes dépendances afin d'en déterminer les éventuelles vulnérabilités puis de les lister sur une page web (figure 7).

En plus de lister les vulnérabilités du projet, l'outil indique diverses informations telles que le nombre d'occurrence de la dépendance dans le projet ou bien la sévérité de la vulnérabilité (High, Medium ou Low).

Pour mettre en place cet outil, j'ai utilisé l'outil Jenkins. Jenkins est un outil d'intégration continu qui permet d'effectuer des actions telles que des

déploiements de projets Maven, exécuter des lignes de commandes, transférer des fichiers,...

Dans la configuration Jenkins, j'ai intégré un script qui permet d'effectuer la recherche de dépendances et de ressortir les résultats sous forme de page html. Seulement, pour que cette page soit intégrée à la documentation des projets Mystique, j'ai intégré à la configuration Maven un transfert de fichier *FTP* pour placer les fichiers concerné dans un endroit spécifique, puis j'ai ajouté des liens sur la documentation pour que le résultat du rapport soit accessible depuis la documentation de Mystique.



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

[How to read the report](#) | [Suppressing false positives](#) | [Getting Help](#): [google group](#) | [github issues](#)

Project: Mystique - Services

Scan Information ([show all](#)):

- dependency-check version: 1.4.4
- Report Generated On: Mar 24, 2017 at 15:04:23 CET
- Dependencies Scanned: 87 (83 unique)
- Vulnerable Dependencies: 10
- Vulnerabilities Found: 157
- Vulnerabilities Suppressed: 0
- ...

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
logback-core-1.1.3.jar	cpe:/a:logback:logback:1.1.3	ch.qos.logback:logback-core:1.1.3	High	1	LOW	16
splunk-logger-1.0.3.jar	cpe:/a:splunk:splunk:1.0.3	com.splunk.logging:splunk-logger:1.0.3	High	7	LOW	11
splunk-1.1.8.jar	cpe:/a:splunk:splunk:1.1.8	com.worldline.cloudwallet.splunk:splunk:1.1.8	High	7	LOW	12
mysql-connector-java-5.1.40.jar	cpe:/a:mysql:mysql:5.1.40	mysql:mysql-connector-java:5.1.40	High	98	HIGHEST	19
activemq-core-5.7.0.jar	cpe:/a:apache:activemq:5.7.0	org.apache.activemq:activemq-core:5.7.0	High	15	HIGHEST	21
kahaadb-5.7.0.jar	cpe:/a:apache:activemq:5.7.0	org.apache.activemq:kahaadb:5.7.0	High	15	HIGHEST	17
activemq-protobuf-1.1.jar	cpe:/a:apache:activemq:1.1	org.apache.activemq.protobuf:activemq-protobuf:1.1	High	11	HIGHEST	16

Figure 7 - Liste des vulnérabilités liées aux dépendances

2.4. Quatrième mission : Batch-launcher

Cette mission ne concerne pas le projet Mystique. Un script permettant d'envoyer automatiquement des mails de rappels est régulièrement lancé. Pour pouvoir uniformiser le lancement des *batch*, la direction technique de Worldline a développé une interface Java permettant de gérer les *batch* (lancement, pause, reprise, arrêt). L'objectif de cette mission était donc de mettre en place un lanceur de *batch* en utilisant l'interface de la direction technique.

Cette mission n'a pas été terminée car après avoir étudié le problème, nous nous sommes rendu compte que le *batch* n'était pas adapté pour être utilisé avec l'interface fourni (impossibilité de mettre en pause ou de récupérer l'état du batch à un instant T).

2.5. Cinquième mission : Refonte du site de Total

Total, souhaitant mettre en place un système de paiement à la pompe avec le smartphone, souhaite effectuer une refonte de son site dédié à la gestion des *wallets* de ses clients.

Comme pour les autres sites de gestion de *wallets*, ce projet s'est basé sur Mystique. Les actions importantes du site étant déjà fourni par Mystique, l'objectif de cette mission était de personnaliser le site pour Total.

Cette mission s'est subdivisée en plusieurs tâches. Chacune correspondant à une partie du site à adapter aux besoins de Total. Au final, j'ai participé à la réalisation de ce site par l'intermédiaire de 11 tâches. Une partie de ces tâches correspondent à des modifications esthétiques et une autre partie des modifications plus fonctionnelles.

Un certain nombre de ces tâches n'étant que des changements mineurs dans l'esthétique de la page tel que des changements de nom ou d'icone, je ne parlerai pas de ces tâches.

Une des tâches que j'ai réalisées sur cette mission a été de mettre en place différents profils de connexion ayant chacun des droits différents.

Lorsque l'on se connecte sur Mystique, le site nous redirige sur un service d'authentification spécifique à Worldline. Une fois les bons identifiants insérés et la connexion effectuée, ce service nous propose de choisir un profil afin de pouvoir naviguer sur le site. Les profils diffèrent au niveau des droits que ceux-ci possèdent sur le site. Par exemple, il est possible d'effectuer des actions destructives avec certains profils mais pas avec d'autres. Total souhaitait que trois profils soit disponibles lors de la connexion au site : CREATOR_TOTAL, EDITOR_TOTAL et VIEWER_TOTAL (figure 8). J'ai donc créé ces nouveaux profils et j'ai attribué à chacun d'entre eux les droits que Total voulait qu'ils possèdent. L'attribution des droits avec Mystique fonctionne grâce à un fichier de propriétés contenant les différents profils. Chaque profil se voit attribuer des droits et lorsqu'une personne se connecte avec un certain profil, le site affichera ou non les actions possibles de faire en fonction des droits inscrits dans le fichier de propriétés.

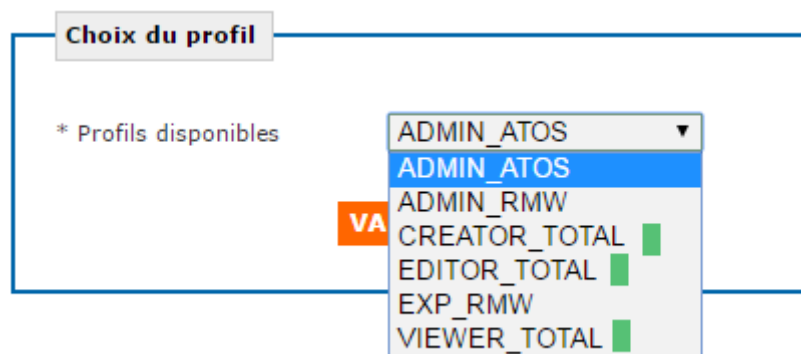


Figure 8 - Sélection du profil utilisateur

2.6. Sixième mission : Génération automatique de documentation

Mystique possède une *API* qui permet au côté *frontend* du site d'utiliser les URL fournis par l'*API* afin d'effectuer différentes actions telles que l'ajout, la suppression, la récupération de données...

Cette *API* possède de nombreuses méthodes et il est bon d'avoir une documentation qui puisse facilement expliquer l'utilité et le fonctionnement de chaque méthode.

Pour réaliser cette documentation, j'ai utilisé un outil de génération automatique de documentation nommé « Swagger' ». Cet outil, une fois intégré au projet, permet de générer une page web répertoriant toutes les méthodes présentes dans l'*API* (figure 9).

Mystique - API

default	Show/Hide List Operations Expand Operations
GET	/admin/auth/cas/callback
GET	/admin/auth/cas/logout
GET	/admin/customers/{customerId}/cards/{cardId}/getWallets
GET	/admin/customers/{customerId}/getCards
GET	/admin/customers/{customerId}/getEvents
GET	/admin/customers/{customerId}/getUsers
GET	/admin/customers/{customerId}/messages
POST	/admin/customers/{customerId}/messages
DELETE	/admin/customers/{customerId}/messages/{id}
PUT	/admin/customers/{customerId}/messages/{id}
GET	/admin/customers/{customerId}/users/{userId}/getWallets
GET	/admin/customers/{customerId}/wallets/{walletId}
GET	/auth/customers/{customerId}/getLoggedInUserInfo

Figure 9 - Documentation sur l'API

Pour chacune de ces méthodes, une vue détaillée est disponible, montrant les paramètres spécifiés pour utiliser la méthode ainsi que le format et les données qui sont retournés par cette méthode (figure 10).

Cette génération s'effectue via le plugin Maven « jaxrs-analyzer ». Ce plugin va s'occuper de chercher toutes les *annotations* dans l'*API* puis générera le site en fonction des résultats que le plugin a récupérés. Une fois la génération effectuée, j'ai intégré la documentation de l'*API* dans la documentation de Mystique.

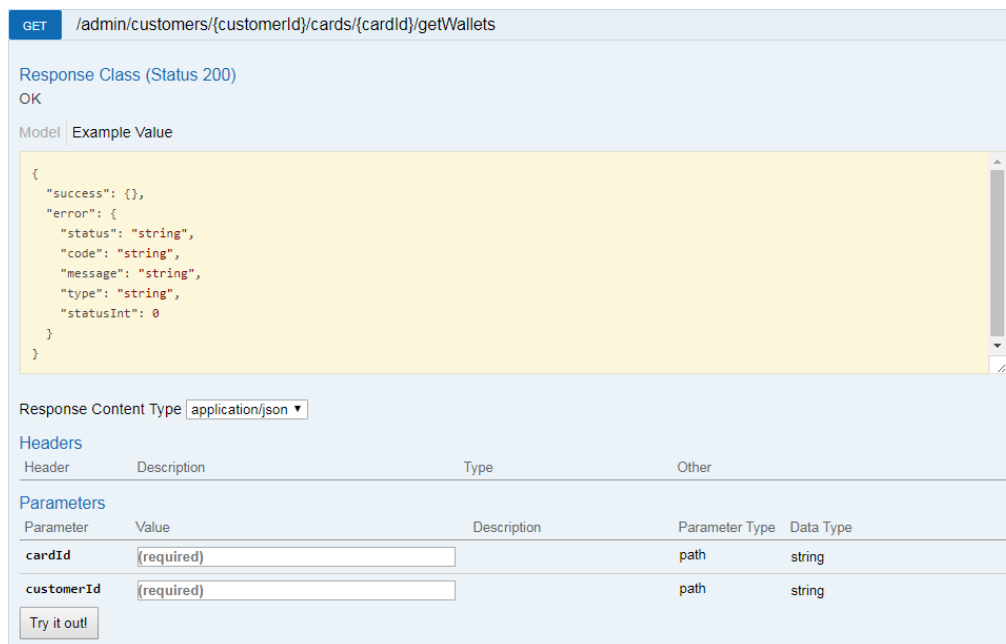


Figure 10 - Détails d'une méthode de l'API

2.7. Septième mission : Extraction de données

Paylib est une solution de paiement permettant de ne pas écrire ses informations bancaires à chaque transaction. Leur site de gestion de *wallet* étant vieux, ils ont décidé d'effectuer une refonte dont notre équipe s'est occupé. Sur cette refonte, je me suis occupé de la page concernant une extraction au format CSV de données stocké en base de données.

La page permet deux extractions différentes : la première concerne des données statistiques sur les *wallets* sur une période de temps donnée et l'autre concerne des informations sur les transactions d'un jour donné.

Pour les statistiques sur les *wallets*, les données sont enregistrées dans une base de données MySQL tandis que les transactions sont enregistrées dans une base MongoDB.

Afin de récupérer les données et exporter dans un fichier CSV, il a fallu mettre en place un cheminement depuis la base de données jusqu'à la vue du site.

Le fonctionnement est le suivant. Lorsqu'une demande d'extraction de données est demandée par la vue du site web, le controller lié à la page, qui s'occupe de gérer les événements sur la page, va appeler une méthode d'un service angularJS permettant de faire des appels http. Un appel http est une requête envoyée par URL et qui permet de récupérer des informations de la part d'une API. Une fois que le service envoie la requête, l'API située côté *backend* va analyser les paramètres spécifiés dans l'URL pour les envoyer au controller métier. Le controller métier est le controller *backend*, et fait le lien entre l'API et le repository qui est l'entité effectuant la requête vers la base de données (BDD). Une fois que le controller a mis en place tous les éléments pour effectuer la récupération des données, il les envoie au repository qui effectue la requête SQL.

Les données ainsi récupérées sont stockées dans des objets java (classes java). Ces objets préalablement créés, sont spécifiés dans des persistence unit. Une persistence unit est une entité listant les objets java liés à une datasource. Une datasource est le lien entre le serveur du site et la base de données.

Pour illustrer l'explication, prenons un exemple. Nous souhaitons récupérer les informations des transactions effectuées le 25 juillet 2017. Le contrôleur angularJS va donc récupérer la date et l'envoyer au service angularJS. Suite à cela, le service va effectuer une requête http avec la date en paramètre. L'API côté serveur va récupérer cette URL, et envoyer la date au contrôleur qui va la transmettre au repository. Le repository va effectuer la requête SQL vers la base de données en utilisant la date du 25 juillet 2017. Pour que cette requête puisse s'exécuter, une datasource contenant les informations de connexion à la base de données sont inscrites dans le fichier de configuration du serveur. Une fois que la requête est correctement exécutée, les informations récupérées vont se stocker dans un objet java correspondant à une transaction et spécifiés dans la persistence unit liée à la datasource. Enfin, les objets Transaction sont renvoyés jusqu'à la vue qui va s'occuper de les traiter et de les transformer en données au format CSV (figure 11).

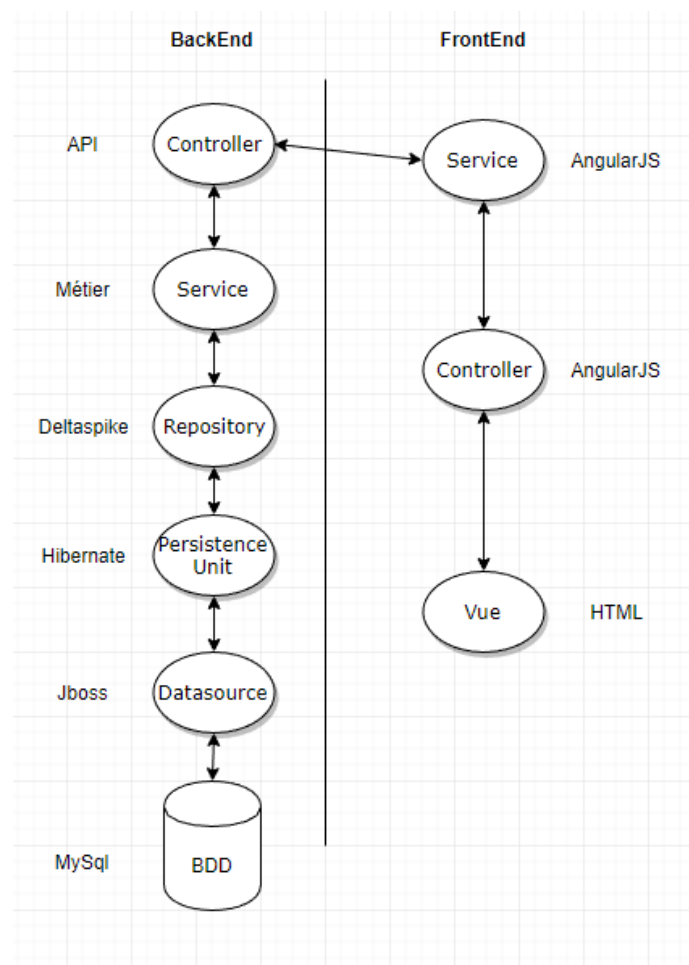


Figure 11 - Récupération des données dans une base MySQL

Le fonctionnement de la récupération des données dans une base mongoDB est un peu plus simple. Le début est identique. Le contrôleur envoie les informations

au service et fait un appel http à l'API du serveur qui envoie les informations au controller du serveur. C'est à ce niveau que la récupération se simplifie. Plus de besoin de persistence unit, datasource et de repository pour récupérer les données. Il suffit d'avoir créé un objet java permettant de récupérer les données. Le controller va utiliser des méthodes permettant de se connecter à la base mongoDB et d'effectuer des requêtes sur cette base pour remplir les objets java avec les informations récupérées avant de renvoyer les objets jusqu'à la vue (figure 12).

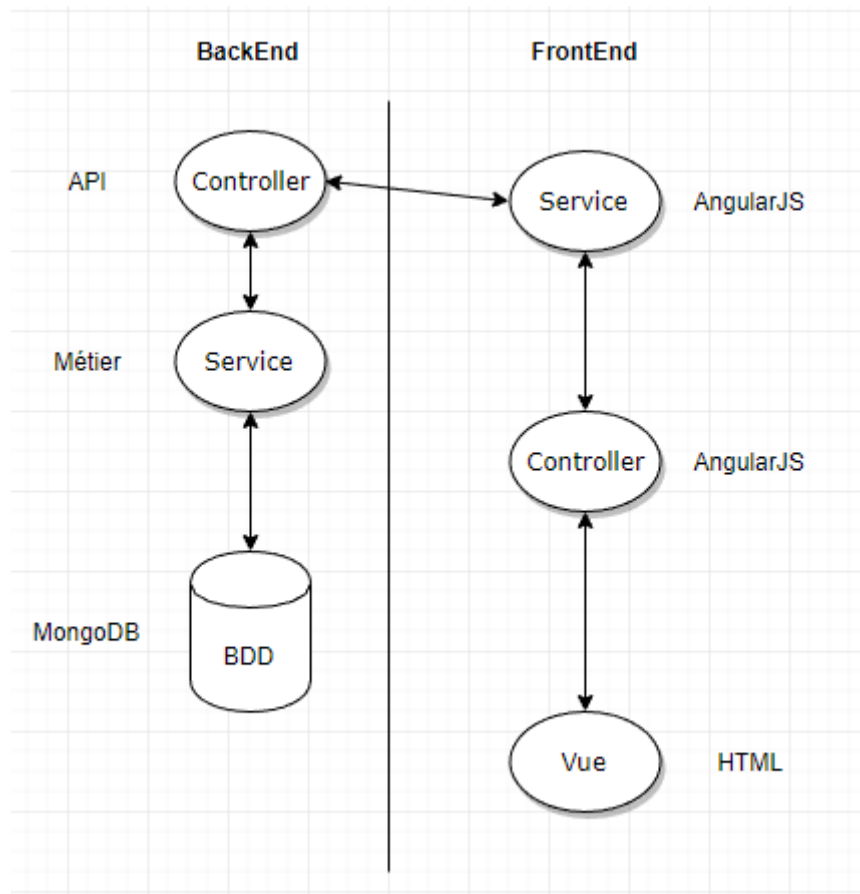


Figure 12 - Récupération des données dans une base mongoDB

3. Evolution des compétences

Durant cette première année, j'ai énormément développé de nouvelles compétences. Au début de ma formation, je n'avais que très peu d'expérience en entreprise, et mes compétences techniques se limitaient à celles acquises par les enseignements de mon IUT. Malgré tout, celles-ci m'ont permises de commencer à travailler au sein de l'équipe de Nicolas NALINE.

Grâce à mon stage de fin d'IUT, j'ai pu découvrir et commencer à utiliser l'outil Maven. Cette compétence m'a été extrêmement utile pour le début de mon apprentissage en entreprise étant donné que mes premières missions se basaient principalement sur cet outil. Cela m'a donc permis de commencer mon travail avec des connaissances existantes et surtout de pouvoir approfondir ma connaissance de l'outil Maven.

Une autre compétence que je tiens de mon IUT et que j'ai pu utiliser durant mes missions en entreprise est le langage Java. Au cours de mon IUT, j'ai appris les bases de ce langage et cela a été utile notamment lorsque j'ai travaillé sur le batch-launcher et l'extraction des données. Cependant, les compétences en Java que je possédais concernaient uniquement le langage Java et non pas le Framework JEE, beaucoup plus complet. Comme Mystique est développé avec JEE, cela m'a permis de monter en compétence sur JEE et d'apprendre à utiliser des technologies comme Hibernate, Deltaspike, etc...

En plus des compétences que j'avais déjà commencé à développer pendant ma formation, j'en ai développé de nouvelles durant mon année en entreprise.

La première est, de par le travail en équipe, l'agilité. L'agilité est un ensemble de méthodes de gestion de projet qui permettent de cadrer la réalisation d'un projet selon une méthodologie spécifique. Sans rentrer dans les détails, les méthodes agiles sont des méthodes de pilotages et de réalisations de projets qui se veulent assez flexibles dans le sens où une acceptation aux changements est possible plutôt qu'une conformité stricte aux plans de base. Le fonctionnement de ces méthodes se base sur des itérations. C'est-à-dire des courtes périodes de développement (2 semaines dans mon cas) au bout desquelles il est possible de visualiser l'avancement du projet avec des exemples fonctionnels de ce qui a été réalisé pendant cette période. Cela permet de juger si le projet avance bien et dans la bonne direction et ainsi prioriser les besoins.

Au sein de l'équipe, cette méthode s'est montrée par la mise en place de petites réunions régulières permettant de voir l'avancer de chaque personne participant au projet. En plus de ces réunions, un outil de gestion et attribution des tâches est mis en place afin de définir le travail à faire, en cours de réalisation et terminé.

Dans un domaine plus technique, j'ai aussi découvert une nouvelle technologie web : l'angularJS. AngularJS est le Framework Javascript utilisé pour Mystique. À mon arrivée dans l'entreprise, je n'avais jamais développé en angularJS et je n'avais même jamais utilisé le langage Javascript. Le fait que les différents projets sur lesquels j'ai pu travailler utilisent ce langage m'a permis de monter en compétences dessus.

En autre compétence technique, il y a mongoDB. MongoDB est un système de gestion de base de données (SGBD) orienté documents. Contrairement aux autres SGBD tels que MySQL, Oracle Database ou encore Microsoft Access, mongoDb ne nécessite pas de schéma prédéfini de données. C'est-à-dire qu'aucune relation n'est présente et mongoDB va se contenter, en fonction des requêtes reçues, de renvoyer les documents enregistrés dans la base. Le travail sur l'exportation des transactions m'a permis de travailler avec ce SGBD et de monter en compétence dessus.

En résumé de l'évolution de mes compétences au cours de cette première année, j'ai approfondi et amélioré des compétences que je possédais déjà (Maven, Java, SQL) mais j'ai aussi appris à utiliser de nouvelles technologies et donc développé de nouvelles compétences (AngularJS, JEE, MongoDB). En plus de ces compétences techniques, j'ai aussi découvert le travail en équipe suivant une méthodologie de gestion de projet agile.

Conclusion

Cette première année d'apprentissage a été très intéressante. J'ai pu découvrir le monde de l'entreprise et le fonctionnement du travail en équipe autour de plusieurs projets partagés par les différents membres de cette équipe. Selon moi, effectuer un apprentissage en alternance apporte beaucoup d'avantages.

Le premier avantage concerne les compétences techniques. Là où les enseignements des écoles apportent beaucoup de connaissances généralistes sur diverses technologies informatiques, travailler sur un projet en entreprise permet d'approfondir des connaissances plus précises (celles utilisées dans les projets) et surtout de découvrir de nouvelles technologies dont on n'entend pas ou très peu parler en cours. Ainsi, cela permet d'élargir notre cercle de compétences techniques et de le renforcer.

Le second avantage est que l'apprentissage au sein d'une entreprise permet de découvrir et d'apprendre comment fonctionne le travail dans le monde professionnel. Le fonctionnement et le déroulement d'un projet est très différent entre les cours et la réalité du travail. Au travail, le projet est beaucoup plus cadré, l'équipe est souvent plus grande et il faut mettre en place des règles et des méthodes afin de mener le projet le mieux possible.

Le dernier avantage est l'expérience. L'alternance permet d'avoir trois années d'expérience en plus par rapport à la formation classique. C'est sûrement l'aspect le plus instructif pour le développement personnel. L'expérience rejoint à la fois la technique et la méthode dans le sens où dans les deux cas, le fait d'avoir pratiqué et d'avoir participé à des projets forge les compétences et permet d'avoir un avis plus critique sur des situations qui peuvent être compliquées pour des personnes n'ayant jamais travaillé en entreprise. Pour prendre un exemple concret, au début de mon apprentissage, j'avais extrêmement de mal à découper des tâches en sous-tâches afin de rendre le travail plus facile à réaliser et à estimer. À force de faire l'exercice de découpage, la tâche devient de plus en plus facile à faire et je vois plus facilement la manière de faire.

Au-delà des avantages techniques et méthodiques apportés par l'alternance, le côté humain est aussi très important. Travailler en entreprise au sein d'une équipe ne signifie pas travailler seul du matin au soir mais échanger avec son équipe, travailler en collaboration afin que le travail soit le plus agréable et le plus productif pour chacun. C'est pour cela que l'intégration dans une équipe est très importante. Une bonne intégration influera grandement sur la productivité et la qualité du travail de la personne. Me concernant, j'ai été très bien intégré dans l'équipe que ce soit par mon manager Nicolas NALINE qui suit de très près ma formation, ou bien par mes collègues de travail qui m'aident beaucoup sur des sujets qu'ils maîtrisent mieux que moi.

Références et sources

<http://fr.atos.net/fr-fr/accueil.html>

<http://worldline.com/fr/accueil.html>

<https://fr.wikipedia.org/wiki/Worldline>

<https://fr.wikipedia.org/wiki/Atos>

[https://fr.wikipedia.org/wiki/Apache Maven](https://fr.wikipedia.org/wiki/Apache_Maven)

Résumé

Résumé :

Au sein de l'entreprise Worldline et travaillant dans l'équipe de Nicolas NALINE dans un cursus d'apprentissage en alternance, ce rapport montre le travail effectué au cours de cette première année ainsi que l'évolution de mes compétences.

Concernant mes missions, cette année ne s'est pas basé autour d'un grand projet mais autour de plusieurs missions. La diversité de ces missions m'a permis de travailler sur plusieurs technologies et par conséquent de monter en compétences sur celles-ci.

Les formations en apprentissage sont un avantage car elles permettent de travailler sur des projets concrets tout en nous permettant de progresser sur des nouvelles technologies ou des méthodologies très demandé dans le monde du travail.

Mots-clefs : Compétences, Worldline, Polytech, Maven, JEE, AngularJS, Agile, Jenkins

Abstract :

Within the Worldline Company and working into Nicolas NALINE's team through an apprenticeship, this rapport shows the work I did during this first year and the development of my skills.

Regarding my missions, this year has not been focused on one big project but on many smaller projects. The diversity of those projects allows me to work on different technologies and consequently increase my skills concerning these technologies.

Apprenticeship's formations are a real advantage because they allow us to work on concrete project all in increasing our skills on new technologies or methodology or work that are a lot seek in the working world.

Keywords : Skills, Worldline, Polytech, Maven, JEE, AngularJS, Agile, Jenkins