

## **Rapport de projet de fin d'année**

### **Développement d'une application interne de gestion des ressources humaines avec Spring Boot / AngularJS**

Présenté par : Hassan Benhzaine

Entreprise : SQLI Rabat

Responsable du stage : Mr. Hicham BENGARAI

Encadré par : Mr. Adnan Ait El Fatmi

Mr. Abdellatif Belhadj

Stage effectué entre le 01/09/2021 et 29/10/2021

# **Sommaire**

## **Chapitre 1: Introduction**

### **1 - Présentation du stage**

### **2 - Présentation de l'entreprise d'accueil**

- Description du groupe SQLI
- Chiffres Clés de SQLI
- La répartition du groupe SQLI

## **Chapitre 2: Problématique**

### **1 - Description de la problématique à traiter**

## **Chapitre 3: Analyse, conception et choix des technologies**

### **1 - Moyens techniques utilisés**

- Outils de développement
- Technologies web / Front-end
- Technologies web / Back-end

### **2 - Analyse fonctionnelle et conception**

- Diagramme de classes
- Architecture de l'application (Back-End)
- Design Pattern
- L'architecture de l'application (Front-end)

### **3 - Gestion et mise en œuvre du projet**

- Tâches du projet
- Démonstration

## **Chapitre 4: Bilan des compétences acquis**

### **1 - AngularJS**

- Le concept de la route
- Le concept de service
- Le service \$http
- Les directives

### **2 - Spring Boot**

- L'auto-configuration
- Annotations
- Autowiring

## **Chapitre 5: Webographie et Bibliographie**

## **Chapitre 6: Remerciements**

# **Introduction**

## **1- Présentation du stage**

Ce rapport de stage prend place dans le cadre de formation en développement Web et Mobile à l'école YouCode. Il présentera le travail réalisé au sein de l'entreprise SQLI au cours du stage. La présentation du travail comporte une partie concernant les résultats obtenus ainsi qu'une partie décrivant la gestion de projet établie.

Au cours de mon stage, je suis intervenu sur un projet. Le projet concerne une application qui permet à l'entreprise d'une part d'impliquer ses collaborateurs dans la gestion des Ressources Humaines avec une approche centralisée et purement administrative : la mise en place de "Manager RH" permet notamment de préserver un management de proximité et ainsi d'assurer un haut niveau d'échange ; et d'autre part de proposer à ses collaborateurs, un environnement et des moyens favorisant leur épanouissement dans leur travail.

Par ailleurs, la consultation des graphes d'évolution ou ce qu'on appelle les "Bilan de Performance", permet au collaborateur d'être véritablement "acteur" de son évaluation et de son évolution au sein de l'entreprise.

Au travers de ce projet, j'ai eu l'opportunité de développer un périmètre complet du logiciel de l'analyse des besoins clients à la conception et au développement.

Le projet n'était pas terminé au moment de l'écriture de ce rapport, le travail présenté couvre l'état actuel de l'avancement du projet.

## 2- Présentation de l'entreprise d'accueil

Ce chapitre décrit brièvement l'organisme d'accueil. Il apporte des précisions sur ses différentes spécialités ainsi que l'ensemble de ses sites.

### Description du groupe SQLI

Créée il y a 10 ans, l'agence accompagne les grandes entreprises et les collectivités dans leur transformation digitale, depuis le conseil métier jusqu'à la mise en œuvre et la maintenance de dispositifs innovants.

Ses équipes s'appuient sur une expertise de pointe des technologies pour :

- Améliorer les ventes de ses clients : marketing digital & social, e-commerce, data marketing, UX et nouveaux usages.
- Renforcer leur performance interne : conception de socles technologiques, Internet, Intranet, conseil et solutions Big Data, solutions mobiles.

### Chiffres Clés de SQLI

- 60% de clients Offshore
- 22 années d'existence
- 18 ans d'expérience des projets e-business
- 50% du chiffre d'affaires au forfait
- 2000 collaborateurs
- Cotée au nouveau marché depuis juillet 2000



## **La répartition du groupe SQLI**

Au niveau du fonctionnement de l'entreprise, il faut noter que le groupe est organisé en agences. Les agences sont découpées en entités de business (Business Unit ou BU). Une BU est un centre de responsabilités dont l'objectif s'exprime sous la forme d'un résultat (marge ou contribution) à obtenir. C'est un département d'entreprise spécialisé. Chez SQLI les BU sont spécialisées dans un domaine d'activité.

Une agence est sous la responsabilité d'un manager d'agence. Une BU est placée sous la responsabilité d'un manager de BU. Chaque collaborateur est rattaché à une et une seule BU.

SQLI est composée de 8 agences en France regroupant 11 sites (Aix en Provence, Belfort, Bordeaux, Dijon, Lyon, Montpellier, Nantes, Paris, Poitiers, Strasbourg, Toulouse), 1 agence au Luxembourg, 2 agences en Suisse (Genève et Lausanne) et 3 agences au Maroc (Casablanca, Rabat et Oujda), ce qui lui permet de bénéficier d'une large couverture géographique, et aussi permet une grande proximité avec les clients et une présence internationale.

Ces agences disposent des forces commerciales et des moyens de production pour la réalisation des projets gagnés en local sur l'offre générique de SQLI et apportent un support aux centres de services spécialisés dont le rôle est de promouvoir une offre au niveau national et international. Ainsi SQLI s'est doté de 5 centres de services spécialisés dans les métiers de la santé, les métiers du Maintien en Condition Opérationnelle et de la Logistique, les logiciels Open Source, le conseil, la formation. SQLI compte plus de 2000 collaborateurs en France et à l'étranger et plus de 800 clients actifs. Les projets sont conduits auprès de grands comptes dans tous les secteurs d'activités.

## **Problématique**

Fort de ses 2000 Collaborateurs à travers ses différentes agences, le groupe SQLI occupe une position centrale dans le marché des NTIC. Cette large base en termes de ressources humaines nécessite une informatisation de l'ensemble des pratiques mises en œuvre pour administrer, gérer et structurer ces ressources impliquées dans l'activité du groupe.

Le challenge consiste à réaliser une application pour la gestion des ressources humaines.

La mise en place de ce projet répond à trois enjeux stratégiques majeurs pour le groupe SQLI:

- Offrir à ses collaborateurs la possibilité de gérer leur carrière.
- Homogénéiser les meilleures pratiques au sein des agences, pour renforcer la qualité des équipes au niveau individuel et collectif.
- Faire du management des Ressources Humaines un critère de différenciation afin d'attirer les meilleurs candidats et fidéliser ses collaborateurs

# **Analyse fonctionnelle, conception et choix des technologies**

## **1 - Moyens techniques utilisés**

### **Outils de développement**

Ayant travaillé sur ce projet au cours de mon stage, j'ai eu l'occasion d'utiliser plusieurs outils de développements dont voici la liste :

- **IDEA IntelliJ**

La partie back-end en java est développée sur l'IDE IntelliJ Idea, cet IDE a été choisi parce qu'il intègre des outils de gestion Git, des outils de gestion Maven et qu'il supporte le framework Spring Boot.

- **Microsoft Visual Studio Code**

Un IDE que pour le développement du module Web App. Il s'agit d'un éditeur de code supportant les langages de programmation Web, comme l'HTML, le CSS, le Javascript et le TypeScript.

- **Postman**

Cet outil me permet de tester les web services ainsi créés. On peut choisir le type de requête HTTP, les données à envoyer et tester la requête avec un affichage du résultat qui en découle. Cet outil garde également en mémoire l'historique des requêtes testées.

- **Google Chrome DevTools**

Un ensemble d'outils de création et de débogage de sites Web intégrés à Google Chrome dans ce projet ces outils sont utilisés pour surveiller les requêtes XHR et la console JavaScript (aider à trouver les erreurs)



- **Angular CLI**

Une Command Line Interface (interface en ligne de commande, en français) développée par les équipes d'Angular même. Cette CLI va faciliter la création des projets dans lesquels la CLI pourra ajouter des fichiers et plus exactement des entités, des interfaces et des composants d'Angular.

## **Technologies web / Front-end**

Dans cette partie les différentes technologies utilisées sur le projet sont présentées.

- **AngularJS**

Un framework développé par Google. Je l'utilise pour le développement de la couche présentation. Il utilise le modèle MVVM (Modèle-Vue-Vue-Modèle) et me permet, en complément du HTML, de créer des pages dynamiques.

## **Technologies web / Back-end**

- **Lombok**

Lombok est une librairie Java s'intégrant automatiquement à un environnement de développement et autres outils de développement afin d'améliorer l'expérience de développement du programmeur Java. Cet outil évite au programmeur de coder les différentes méthodes d'accès, de comparaison, d'écriture et autres méthodes triviales.

- **JPA**

(Java Persistence API) est un standard de la plateforme Java EE gérant la persistance des données. Il me permet la gestion de correspondance entre des objets Java (entity) et une base de données.

- PostgreSQL

Un système de base de données. C'est un outil libre qui se base sur le langage SQL. Je l'ai utilisé car c'est sur ce système que sont stockées les données de l'entreprise SQLI.

- Spring Boot

Un framework qui me permet la mise en place d'application Spring rapidement et facilement. Il se base sur le Framework Spring et permet de s'affranchir de la plupart des configurations de celui-ci à mettre en place pour créer une application.

- Java 8

La dernière version de Java offre notamment la possibilité d'utiliser des lambda expressions, une nouvelle API de gestion de dates et heures ainsi que l'utilisation des "Optionals" qui permettent une meilleure gestion des références null.

- Spring Web MVC

Permet de construire des applications Web en Java. Comme son nom le suggère, il utilise le principe du Modèle/View/Contrôleur (MVC) en association avec le modèle IoC (Inversion of Control) du Spring Framework.

- Maven

Maven est un outil de construction de projets open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il me permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java comme les dépendances.

## **2 - Analyse fonctionnelle et conception**

La réalisation de n'importe quel projet informatique démarre par une cette phase primordiale. Car elle permet de définir l'ensemble des briques constitutives de l'application et leurs interfaces. L'importance de cette phase se manifeste techniquement dans le fait qu'une conception bâclée plantera un projet de manière sûre et certaine. A cet effet, j'ai opté pour UML comme méthode de conception Orientée Objet en se servant de ces diagrammes :

### **Diagramme de classes**

Le diagramme de classe est la description de tout ou d'une partie du système d'une manière abstraite, en termes de classes, de structure et d'associations. Le diagramme de classe est réalisé en utilisant le mapping objet-relationnel ou ORM. Cette méthode crée une correspondance entre la base de données relationnelle et les objets du langage, en associant chaque classe avec une table et chaque attribut de la classe avec un champ de la table.

Le diagramme de classes représenté dans la Figure 0 suivante décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes.

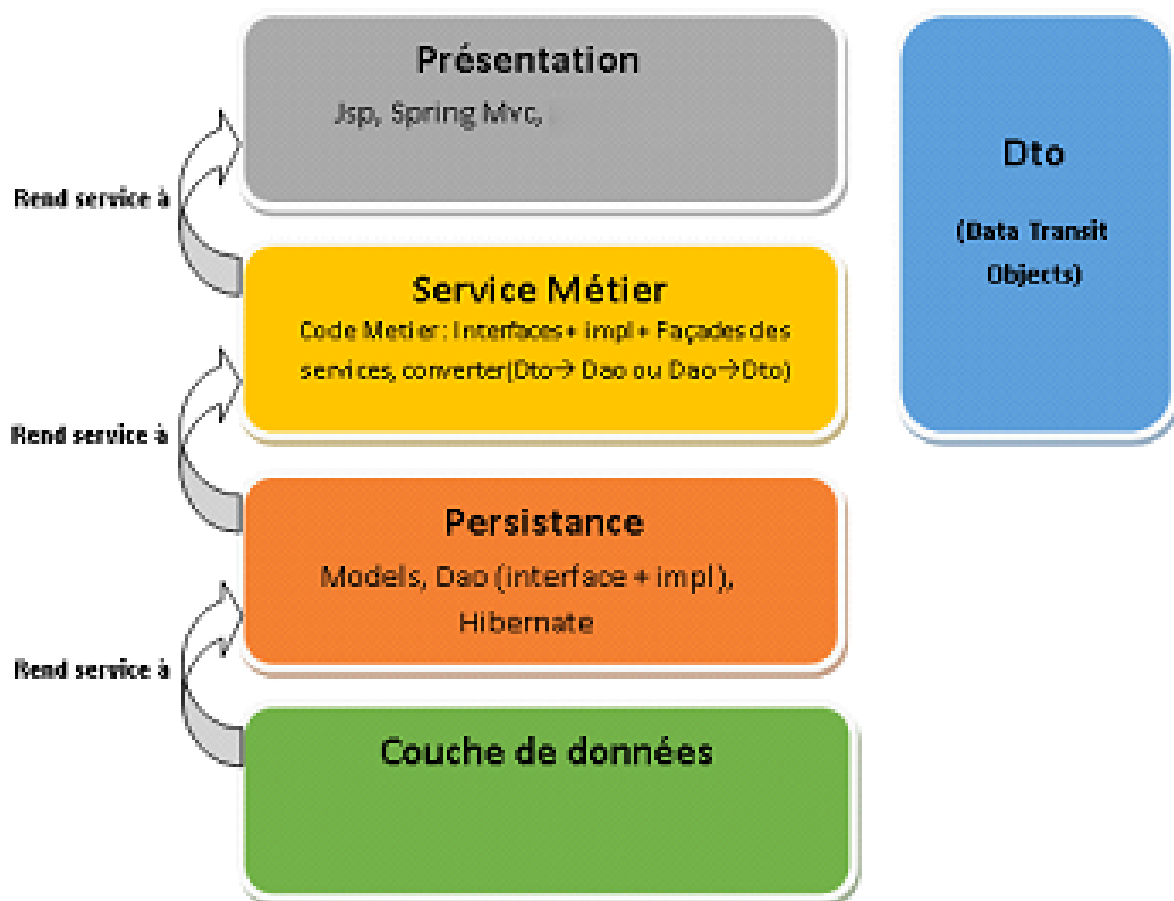


Figure 0: Le diagramme de classe du système d'information

## Architecture de l'application (Back-End)

J'ai avais choisi une structuration de l'application en couches parce qu'elle permet :

- de maîtriser la complexité des applications (développement, échanges entre les applications, interactions entre objets)
- d'améliorer le découplage de l'application
- d'optimiser les temps de développement, en factorisant certaines briques applicatives
- de favoriser la communication : à l'intérieur d'une application, en structurant les échanges entre les différentes couches pour que les changements dans l'implémentation d'une couche n'affectent pas la façon dont elle est utilisée par la couche supérieure.



Dans cette structuration, chaque couche ne communique qu'avec la couche inférieure et elles sont liées par des interfaces :

- La couche contrôleur :

Cette couche contient l'interface homme-machine et gère l'organisation des données à afficher. Elle contient des contrôleurs qui organisent l'enchaînement des pages à afficher, en fonction des différentes demandes formulées par l'utilisateur.

- La couche service :

Cette couche rend service à la couche présentation, chaque demande de l'utilisateur correspond à un service appelé par cette couche, qui appelle la couche inférieure (couche d'accès aux données) et renvoie le résultat de son traitement à la couche supérieure.

- La couche accès aux données :

Cette couche prend en charge l'interface avec la base de données. On y trouve les objets qui permettent d'accéder à ces données (les DAOs: Data Access Object).

## **Design Pattern**

Le Pattern vient avec l'idée d'encapsuler dans une couche spécifique tous les mécanismes d'accès aux entrepôts persistants de données :

- Bases de données relationnelles
- Système de fichiers local ou distant
- Requêtes XML-RPC (SOAP ou REST) sur des systèmes tiers

Ce pattern concerne donc la Couche Persistance.

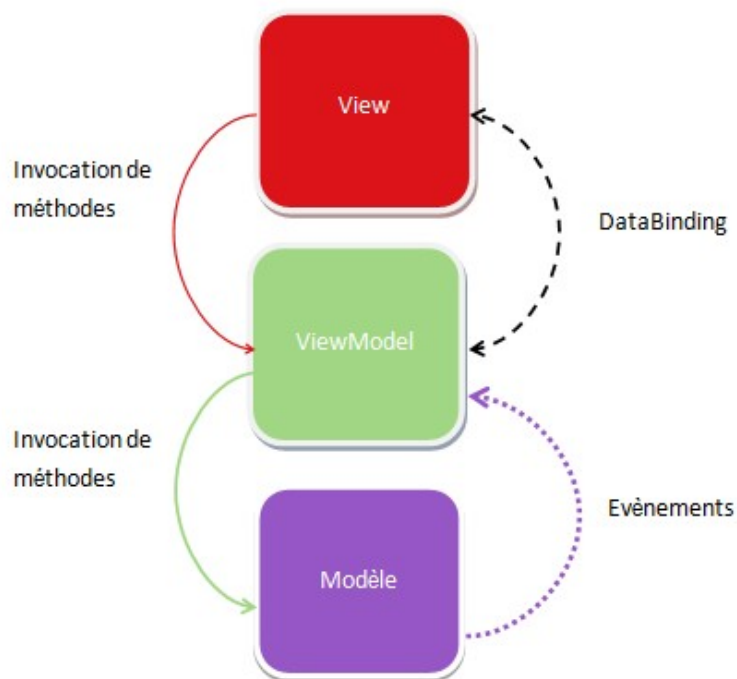
Les DAOs (couche Données Métier) seront donc injectées dans les services (couche Service Métier).

## L'architecture de l'application (Front-end)

Le Modèle-Vue-ViewModel (en abrégé MVVM, de l'anglais Model View ViewModel) est une architecture et une méthode de conception utilisée dans le génie logiciel. MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Presentation Foundation et Silverlight via l'outil MVVM Light par exemple.

Cette méthode permet, tel le modèle MVC (Modèle-Vue-Contrôleur), de séparer la vue de la logique et de l'accès aux données en accentuant les principes de binding et d'évènement. »  
([source](#))

Pour vulgariser, la différence entre MVVM et MVC tient du fait que le MVVM, lui, se repose sur le DataBinding et l'évènementiel. Ça tombe bien, c'est ce que AngularJS est !



La vue est couplée aux données via du DataBinding et invoque les méthodes du ViewModel.

Le ViewModel invoque les méthodes du modèle. Il contient la data spécifique à la gestion de l'écran et les méthodes de réponses aux interactions utilisateurs. Il contient également une référence vers un ou des modèles.

Le modèle contient la data et les méthodes de manipulation de cette dernière (calculs, appels de services, ...).



## **Gestion et mise en œuvre du projet**

La gestion de projet qui est abordée dans cette partie concerne majoritairement la mission principale pour le stage.

### **1 - Tâches du projet**

Cette partie décrit les tâches importantes définies lors du projet et répondant aux problématiques présentées.

- **Formation (tâche 1) 3 semaines**

Cette tâche a pour but de me former aux technologies et outils utilisés. L'objectif est donc de développer une petite application en AngularJS et Spring Boot.

La formation à Java, Spring Boot et AngularJS se traduit par de la lecture de documentation et tutoriels pour comprendre le principe de fonctionnement.

A la suite de ces formations, la conception et la réalisation de l'application a commencé à partir des concepts étudiés.

- **Conception (tâche 2) une semaine**

Cette tâche consiste à enrichir la description du procédé, de détails d'implémentation afin d'aboutir à une description très proche de l'application en modélisant toute l'architecture en diagramme de classes.

- **Réalisation du projet (tâche 3) un mois**

La réalisation est la tâche la plus importante après celle de la conception. Le choix des outils de développement influence énormément la flexibilité du produit à réaliser.

Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système.

## 2 - Démonstration

- **Module de gestion des managers RH**

Avant de manipuler n'importe quelle fonctionnalité de l'application il faut d'abord ajouter un ou plusieurs managers RH en utilisant l'interface d'ajoute

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

firstName

lastName

status

add hr manager

**Figure 1: l'ajoute d'un manager RH**

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

id	status	firstName	lastName		
2	false	ahmad	Hurley	X	<a href="#">edit</a>
3	true	Jack	Foley	X	<a href="#">edit</a>
4	true	Jena	Solomon	X	<a href="#">edit</a>

**Figure 2: gestion des managers RH**

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

firstName

lastName

status

save changes

cancel

**Figure 3: modification d'un manager RH**

- **Module de gestion des compétences**

Ce module permet la gestion des compétences dans l'application pour les assigner ultérieurement dans l'interface d'ajouter d'un collaborator

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

skill name

Java

add skill

**Figure 4: l'ajoute d'une compétence**

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

id	name		
5	C#	<a href="#">X</a>	<a href="#">edit</a>
6	java	<a href="#">X</a>	<a href="#">edit</a>
7	mysql	<a href="#">X</a>	<a href="#">edit</a>

**Figure 5: gestion des compétences**

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

name

java/J2EE

save changes

cancel

**Figure 6: modification d'une compétence**

- **Module de gestion des collaborateurs**

Le derri re module est maintenant 100% fonctionnelle on peut ajouter un collaborateur sans probl me

add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

number  
AD879

firstName  
Omar

lastName  
Stout

sex  
Male

hrManager  
Jena Solomon

site  
Rabat

bu (exemple: MRR1, MRR2)  
MRR3

hiringDate  
2021-10-21

poste APP (exemple: CM2, RE0, ...)  
CM1

Poste Actuel (exemple: ICD1, DEV4, ET3, CPT2, ...)  
CPT2

integrationSeminarDate (OPTIONAL)  
yyyy-mm-dd

salaire  
9000

skills  
☒ C# ☐ java ☒ mysql

Ajouter un collaborateur

**Figure 7: l'ajoute d'une comp tence**

Add collaborator	view collaborators	view hr managers	add hr manager	view skills	add skill
------------------	--------------------	------------------	----------------	-------------	-----------

id	number	first Name	last Name	hr manager	oldHr manager	abbr collab	sex	site	bu	hiring Date	bap Month	app Position	integration Seminar	integration SeminarDate	current Position	salary	
8	32343	OMAR	SCOUT	Ahmad Scout	null	Osc	M	RABAT	MRR1	2021-03-12	MARCH	RE0	true	2021-03-12	DEV4	8000	X

**Figure 8: gestion des collaborateurs**

## **Bilan des compétences acquis**

### **1- AngularJS**

#### **Le concept de la route**

AngularJS encourage les applications « single page », les routes permettent malgré tout de créer des URL qui sont « user friendly », l'utilisateur peut ainsi instantanément savoir sur quelle page il se trouve en se référant à l'URL. Plutôt que d'avoir un URL constant du type :

- example.com

Il devient alors possible d'avoir un URL spécifique pour les pages atteintes

- example.com/contact

L'index de l'application est un modèle de présentation dans lequel on affiche nos vues grâce aux routes que l'on aura définies.

Enfin, les routes permettent de définir un contrôleur propre à chaque page, ce qui évite les contrôleurs volumineux en ligne de codes et permet de séparer les tâches en fonction de la route sur laquelle l'utilisateur se trouve.

#### **Le concept de service**

Le développeur peut créer ses propres services et les appeler dès que nécessaire, comme il le ferait pour des fonctions JavaScript. Ceci s'avère très utile pour des tâches redondantes à plusieurs contrôleurs. En effet, avec les routes, les contrôleurs sont séparés, un service sera donc plus efficace que de copier/coller la fonction que l'on a créé dans un contrôleur précédent. Un service peut être considéré comme une fonction globale.

#### **Le service \$http**

Le Framework met à disposition le service \$http qui permet un accès serveur qui engendre une réponse de type success si la requête a abouti ou erreur si c'est le contraire, comme en AJAX. Le service \$http permet de créer un client RESTful et met donc à disposition les méthodes REST, à savoir :

- \$http.get

- \$http.post
- \$http.put
- \$http.delete

Ce sont les méthodes les plus utilisées. Grâce à ce service, les applications peuvent mettre en place un CRUD complet avec un serveur distant.

### **Les directives**

Les directives permettent d'attacher des comportements spécifiques à des éléments HTML de notre page. Concrètement, cela ressemble un peu à des listeners en Java pour rendre notre page interactive.

Avec tous ces concepts et après avoir suivi les tutoriaux cités précédemment, je peux maintenant développer mon prototype d'application de gestion développé avec AngularJS

## **2 - Spring Boot**

### **L'auto-configuration**

Cette fonctionnalité est la plus importante de Spring Boot. Elle permet de configurer automatiquement votre application à partir des *JAR* trouvés dans votre Classpath. En d'autres termes, si vous avez importé des dépendances, Spring Boot ira consulter cette liste puis produira la configuration nécessaire pour que tout fonctionne correctement.

Prenons l'exemple d'une application web dans laquelle vous avez les dépendances : Hibernate et Spring MVC. Normalement, vous devez créer les fichiers de configuration suivants :

- appconfig-mvc.xml
- web.xml
- persitence.xml

## **Le concept des Annotations**

Au cœur de Spring Boot se trouvent les annotations qui servent à simplifier la tâche au développeur. Pour créer un contrôleur, il suffit de créer une classe et de l'annoter `@RestController` et de lui affecter un point d'accès. Chacune des méthodes aura l'annotation `@RequestMapping` qui indique quel chemin de l'API la méthode couvre et quelle méthode HTTP lui correspond.

Ces annotations permettent de simplifier le code et de le rendre plus lisible. Le framework s'occupe de démarrer le serveur web et de rediriger les requêtes aux méthodes concernées.

## **Le concept d'Autowiring**

Le principe d'autowiring est un autre concept important dans Spring Boot. Simplement en ajoutant l'annotation `@Autowired` au constructeur, Spring s'occupera d'instancier la classe directement en injectant les dépendances décrites dans les paramètres du constructeur, par exemple des services, repositories ou encore des objets de contexte. Pour chacun des objets injectés, il s'agira d'un c, c'est-à-dire que chacune des classes qui ont le même service comme dépendance recevront la même instance du service.

## Conclusion et perspective

Pour mettre en œuvre ce projet, j'ai amené, dans un premier lieu, à établir une étude conceptuelle du sujet afin de dégager les différents modules de cette application ainsi qu'une étude des outils et technologies susceptibles de convenir à sa réalisation. Dans un second lieu, j'ai fait une analyse et conception du projet en se basant sur le formalisme UML. Un diagramme a été élaboré afin de mieux découper le projet, ce qui a facilité sa mise en œuvre. Finalement, j'ai implémenté les différents modules de cette application. Le résultat de cette dernière phase répond aux exigences et aux besoins du cahier de charge.

Au cours de ce projet, j'ai eu l'opportunité de mettre en exercice, différentes connaissances acquises durant mon processus scolaire. De plus, j'ai pu aussi renforcer mes connaissances étudier au sein de YouCode concernant différents concepts et à savoir la technologie Java, TypeScript ainsi que les différents Framework qui lui sont associés et l'architecture multi-tiers.

Tout au long du travail, j'ai rencontré pas mal de difficultés, notamment le choix des Frameworks front-end car il y'en a beaucoup, ainsi le manque d'expérience en ce genre d'application a nécessité une période intense d'auto-apprentissage. Mais à force de persévérance tout marche correctement.



## Webographie et Bibliographie

- <https://openclassrooms.com/fr/courses/4975451-demarrez-votre-projet-avec-java>
- <https://openclassrooms.com/fr/courses/4989236-apprenez-l-objet-avec-java>
- <https://openclassrooms.com/fr/courses/4504771-simplifiez-le-developpement-dapplicatio>  
[ns-java-avec-spring](https://openclassrooms.com/fr/courses/4504771-simplifiez-le-developpement-dapplicatio)
- <https://openclassrooms.com/fr/courses/6900101-creez-une-application-java-avec-spring-b>  
[oot/exercises/3979](https://openclassrooms.com/fr/courses/6900101-creez-une-application-java-avec-spring-b)
- <https://angular.io/docs>
- <https://stackoverflow.com/>
- <https://www.youtube.com/channel/UCsBjURrPoezykLs9EqgamOA> (Angular)
- <https://www.youtube.com/c/JavaBrainsChannel> (Java tutoriels)
- <https://www.youtube.com/user/djdjalas> (Java tutoriels)
- <https://www.javatpoint.com/> (Documentation Java)
- <https://www.jetbrains.com/help/> (Documentation IntelliJ)
- <https://www.w3schools.com/>

## Remerciements

Je tiens à remercier les personnes qui m'ont accompagné tout au long de mon stage,

Au sein de l'entreprise SQLI, je tiens à remercier M. Abdellatif Belhadj pour m'avoir accompagné tout au long du stage et m'avoir aidé pendant mes développements et répondre rapidement à toute mes interrogations. Je tiens également à remercier toute l'équipe de développement SQLI pour m'avoir accueilli et accompagné tout au long du stage. Je tiens à remercier mes responsables M. Hicham BENGARAI et M. Adnan Ait El FATMI pour leurs accompagnement tout au long du stage.

**Autre**

**Dépot GitHub**

[https://github.com/hassanbenhzaine/stage\\_SQLI\\_2021](https://github.com/hassanbenhzaine/stage_SQLI_2021)