OC Pizza

Mise en place d'un système informatique sur-mesure déployé dans toutes les pizzerias

Dossier de conception technique

Version 1.1.1

Auteur
Florent Ros
Analyste Programmeur FlexThings

FlexThings
www.flexthings.fr
22 Avenue du Maréchal Foch 78300 Rambouillet – 06 19 78 66 65 – florent.ros@flexthings.com
S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de 8965 – SIREN 120 889 297 – Code APE : 6202A

TABLE DES MATIÈRES

1 - VERSIONS

Auteur	Date	Description	Version
Florent Ros	25/11/2021	Création du document	1.1.1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza.

L'objectif du document est d'identifier les besoins techniques de OC Pizza. En effet le commanditaire du projet Lola co-fondatrice de OC Pizza a exprimé les besoins suivants :

Le commanditaire du projet Lola co-fondatrice d'un groupe de pizzaria a exprimé les besoins suivants :

● être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation
● suivre en temps réel les commandes passées, en préparation et en livraison
● suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées
● proposer un site Internet pour que les clients puissent :
- passer leurs commandes, en plus de la prise de commande par téléphone ou sur place
- payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison
- modifier ou annuler leur commande tant que celle-ci n'a pas été préparée
● proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza

L'ensemble de ces fonctionnalités sont prise en compte dans le nouveau système informatique pour l'ensemble des pizzerias du groupe.

A travers ce document nous aurons un idée precise de :

- Les composants propres à la solution ainsi que les composants extérieurs et l'ensemble des interactions
- Le déploiement de la solution proposée
- L'architecture logicielle
- Le Modèle Physique de Données avec le script SQL

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants:

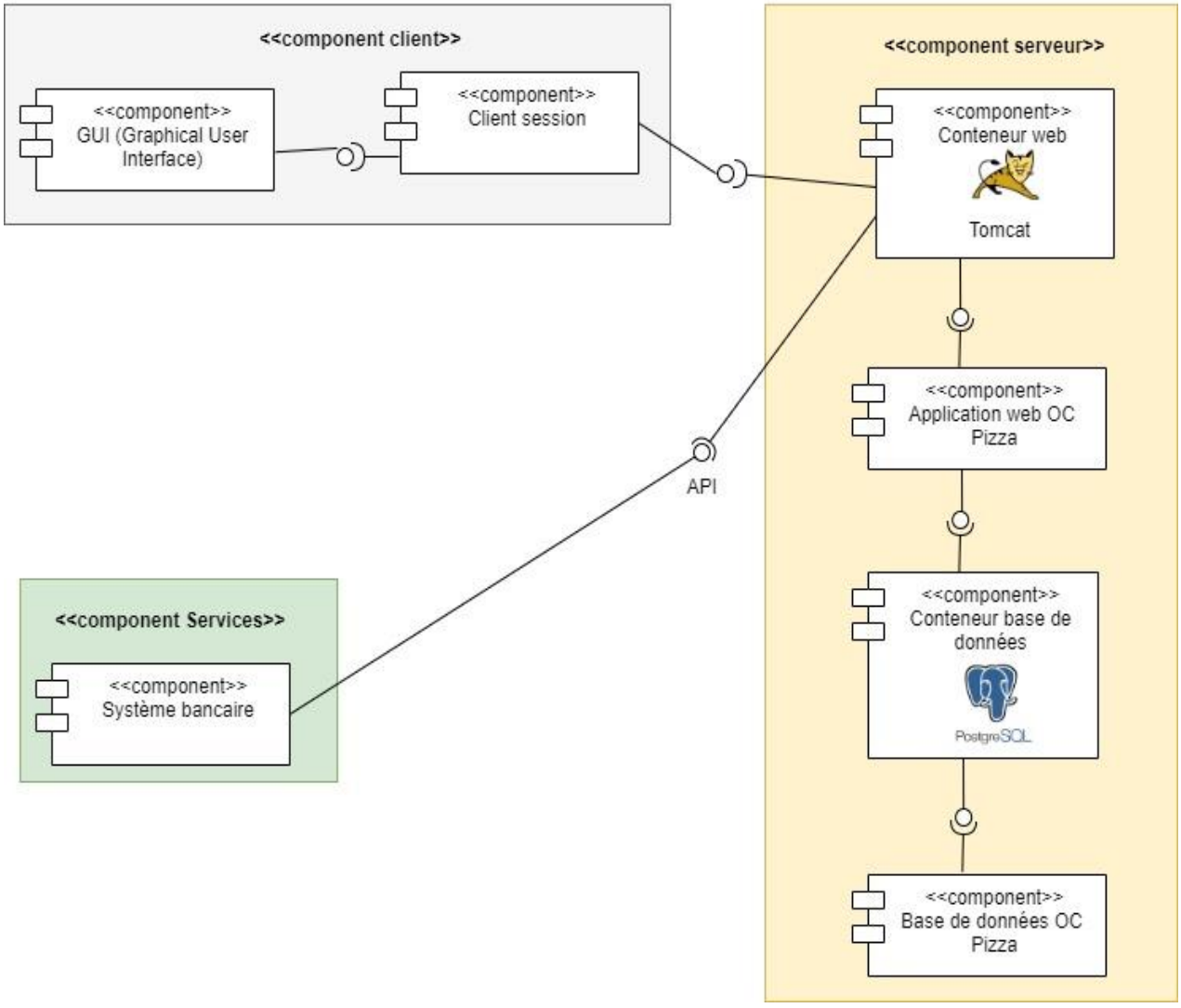
1. **Projet OC Pizza** : Dossier de conception fonctionnelle de l'application
2. **Projet OC Pizza** : Dossier d'exploitation

3 - Architecture Technique

3.1 - Composants généraux

Le diagramme de composants permet d'obtenir une vision d'ensemble du système illustré sous forme de composants. Cela apporte également les relations de dépendance entre des différents composants.

Diagramme de composants



3.1.1 - Component client

Ce composant permet de rechercher les produits proposés et d'afficher leurs informations. De plus il affiche sur l'interface les produits ainsi que leurs informations en attente de commande. Enfin il affiche le montant total de la commande effectuée par le client.

3.1.2 - Component serveur

Le composant base de donnée comporte le stock des ingrédients et des pizzas classes par nom et quantité. Ce composant affiche également le statue des commandes (validé, archive, payé). De plus avec le composant application Web OC pizza également nous affichons toutes les commandes en cours. Par ailleurs ce composant permet d'authentifier les utilisateurs afin de connaitre les droits associés. Enfin nous pouvons aussi gerer les comptes utilisateurs creation, suppression.

3.1.3 - Component services

Le composant système bancaire est un composant extérieur au système, il est relié au component serveur via une API. Ce composant offre la capacité d'accepter les paiements par carte bancaire.

3.2 - API Web

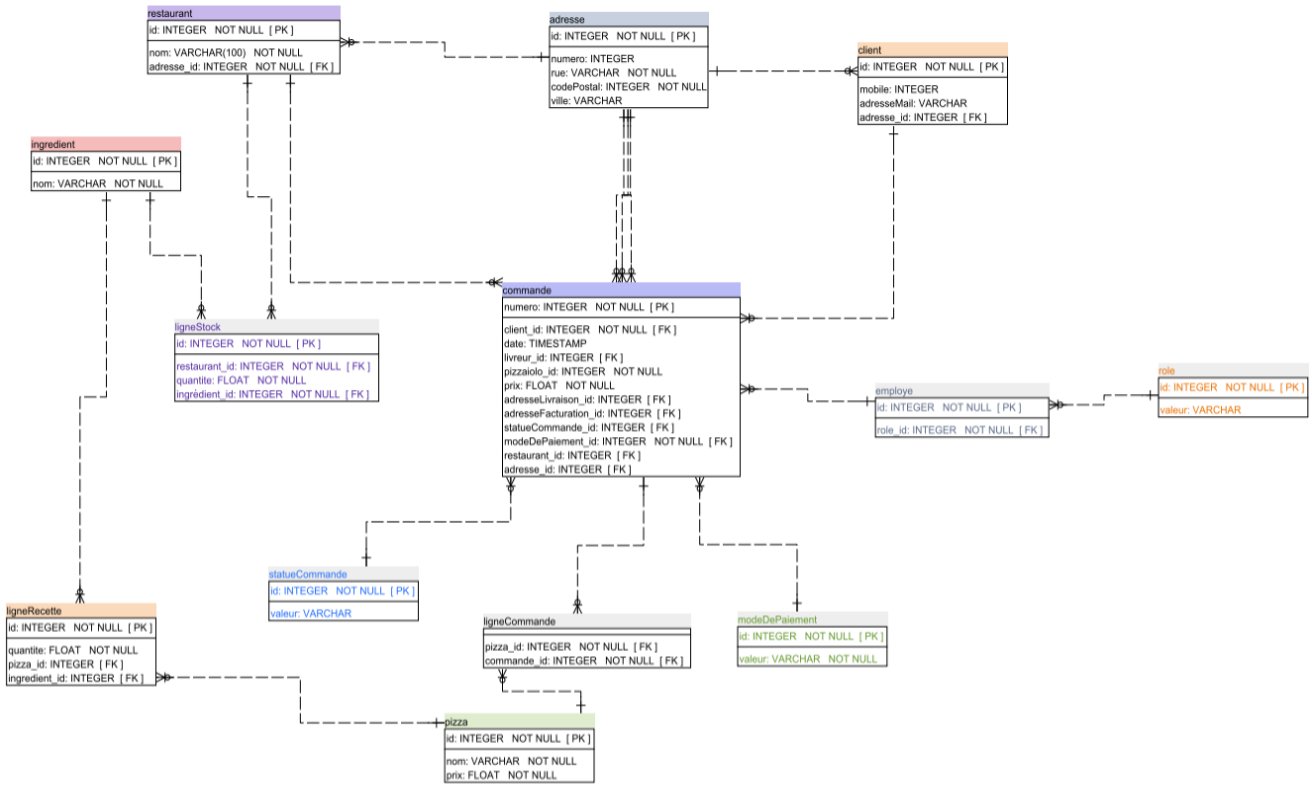
La pile logicielle est la suivante :

- Application JEE (JDK version 1.8)
- Serveur d'application Tomcat 5.2.9
- Base de données PostgreSQL
- Framework Spring

3.2.1 - Modèle physique de données avec script SQL

Le modèle physique de données créer une abstraction de la base de données et génére des schémas. Les types d'entités sont représentés par des tables, et les lignes de type relation représentent les clés étrangères entre les tables. Cela offer une garantie que les objets de données et les relations représentés sont exacts et compatibles avec les systèmes de l'organisation.

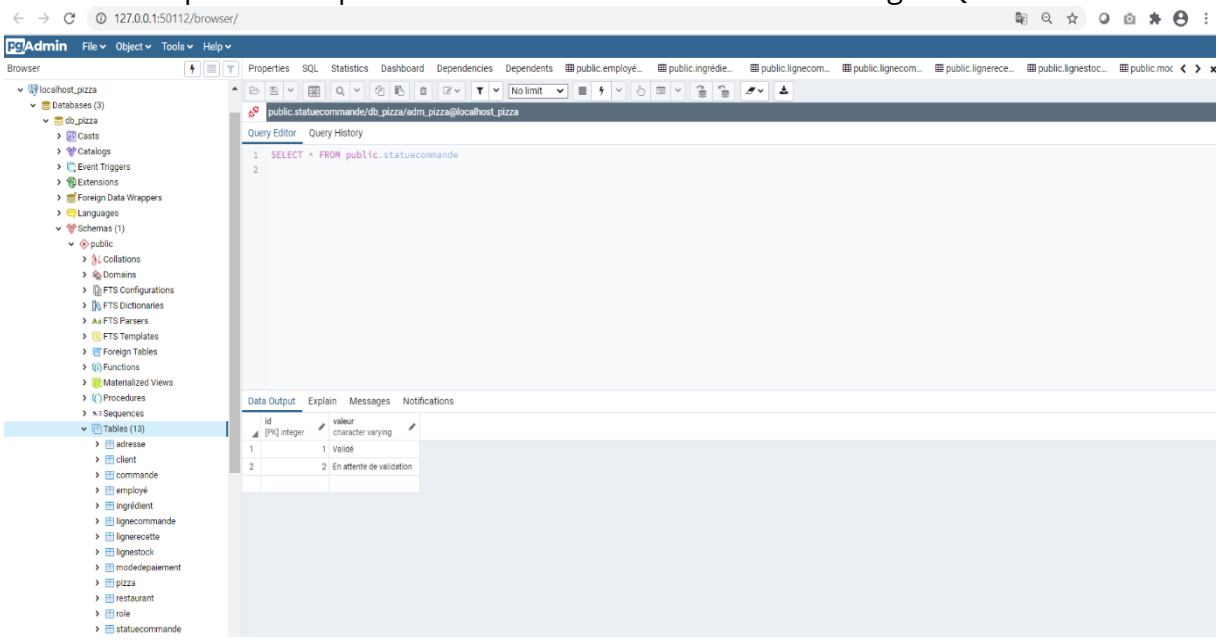
Modèle physique de données



Création du script SQL et integration dans PostgreSQL.
Script SQL :

```
CREATE SEQUENCE public.role_id_seq;

CREATE TABLE public.role (
                id INTEGER NOT NULL DEFAULT nextval('public.role_id_seq'),
                valeur VARCHAR,
                CONSTRAINT role_pk PRIMARY KEY (id)
);

ALTER SEQUENCE public.role_id_seq OWNED BY public.role.id;

CREATE SEQUENCE public.employ__id_seq;

CREATE TABLE public.employé (
                id VARCHAR NOT NULL DEFAULT nextval('public.employ__id_seq'),
                role_id INTEGER,
                CONSTRAINT employ__pk PRIMARY KEY (id)
);

ALTER SEQUENCE public.employ__id_seq OWNED BY public.employé.id;

CREATE TABLE public.pizza (
                id INTEGER NOT NULL,
                nom VARCHAR NOT NULL,
                prix REAL NOT NULL,
                CONSTRAINT pizza_pk PRIMARY KEY (id)
);

CREATE SEQUENCE public.statuecommande_id_seq;

CREATE TABLE public.StatueCommande (
                id INTEGER NOT NULL DEFAULT nextval('public.statuecommande_id_seq'),
                valeur VARCHAR,
                CONSTRAINT statuecommande_pk PRIMARY KEY (id)
);

ALTER SEQUENCE public.statuecommande_id_seq OWNED BY public.StatueCommande.id;
```

Tables correspondantes présentes dans la base de données PostgreSQL :



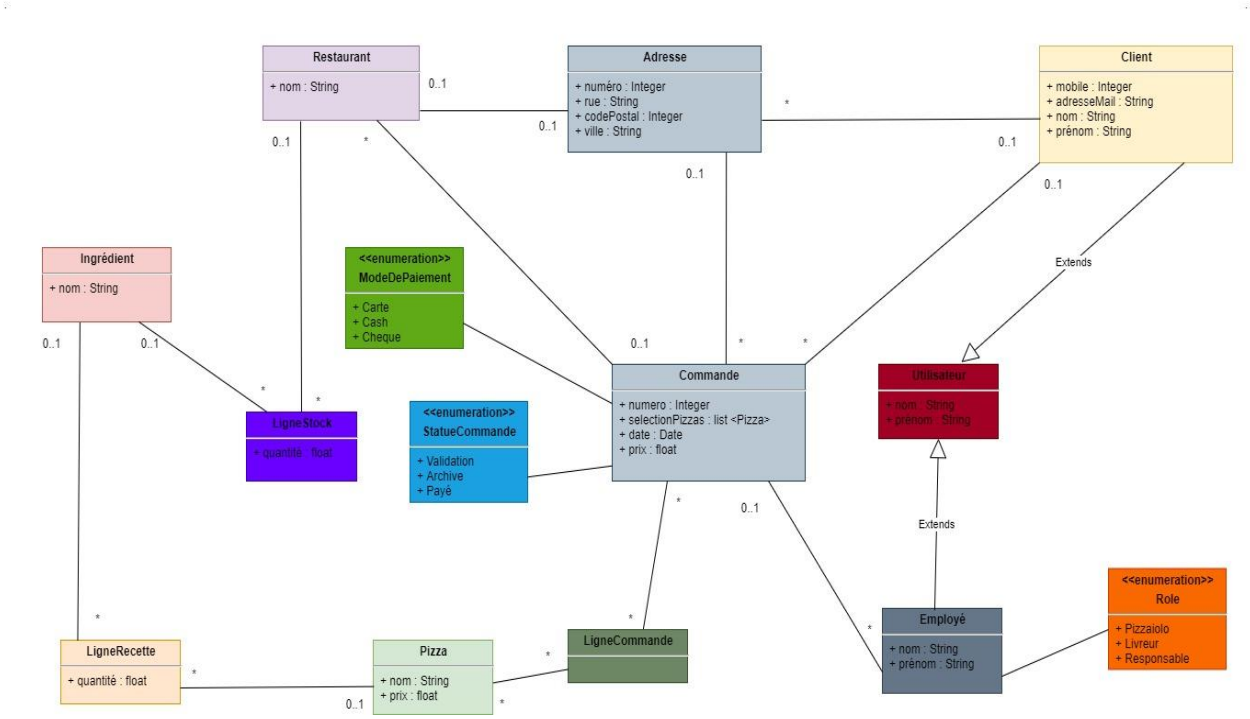
3.3 - Application Web

La pile logicielle est la suivante :

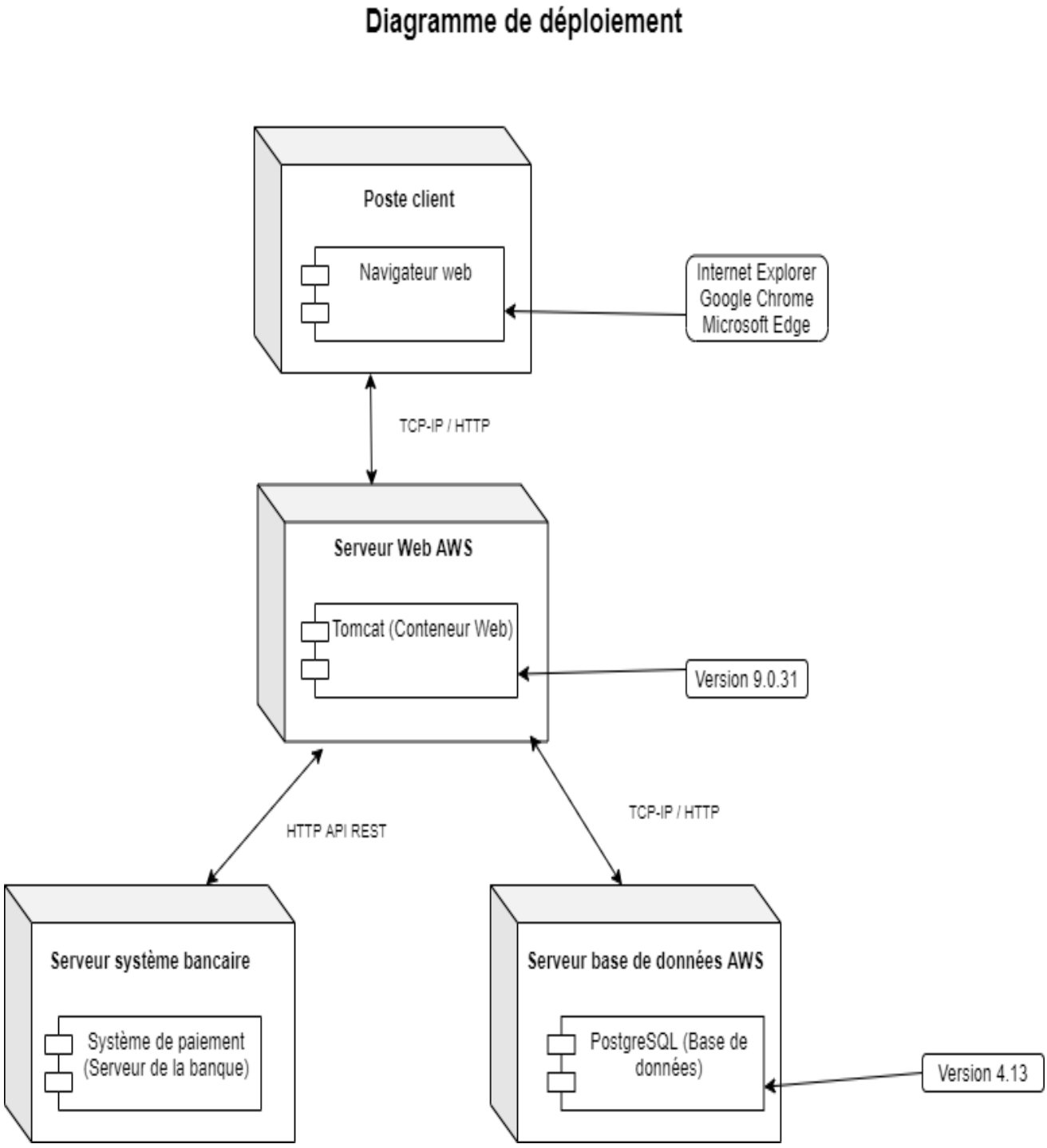
- Application JEE (JDK version 1.8)
- Serveur d'application Tomcat 5.2.9
- Framework Angular

3.4 - Diagramme de classe du domaine fonctionnel

Diagramme de classes du domaine fonctionnel



4 - Architecture de Déploiement



4.1 - Serveur de Base de données

La base de données sera déployée sur le serveur dédié où sera aussi installé le serveur d'application Apache Tomcat pour l'API Web 9.0.31.

Caractéristiques techniques Serveur Linux Debian Jessie + PostgreSQL 4.13

4.2 - Serveur dédié API Web

Le déploiement des applications web nécessitent un serveur dédié capable de prendre en charge un trafic important, c'est la raison pour laquelle un serveur dédié sera loué auprès d'AWS.
Caractéristiques techniques Serveur AWS instance ec2 + Apache Tomcat 9.0.31

4.3 - Serveur dédié Web application

Un second serveur dédié sera loué auprès de AWS afin de déployer la partie Frontend de notre application.
Caractéristiques techniques Serveur AWS instance ec2 + Apache Tomcat 9.0.31

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

Les sources et versions du projet sont gérées par Git, les dépendances et le packaging par Apache Maven.

Le développement de l'application est basé sur les Framework Spring Boot et Angular. Il est divisé en deux parties web application basée sur le Framework Angular (frontend) et une API Web en REST basée sur le Framework Spring Boot (backend).

5.1.1 - Les couches

L'architecture applicative pour l'API est la suivante :

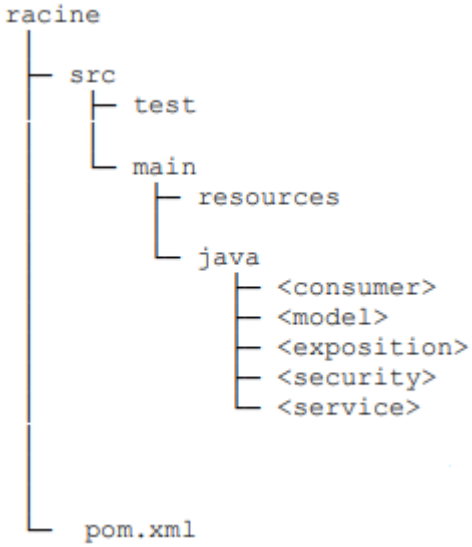
- Une couche service : responsable de la logique métier du composant
- Une couche consumer : qui correspond à la couche d'accès aux données
- Une couche model : implémentation du modèle des objets métiers
- Une couche controller : gère les requêtes client. Point d'entrée de l'application
- Une couche security : Gère l'authentification des utilisateurs et la vérification des droits utilisateurs

L'architecture applicative pour la web application est la suivante :
- Une couche service : responsable de la logique métier du composant
- Une couche model : implémentation du modèle des objets métiers
- Une couche consumer : qui correspond à la couche d'accès aux données
- Une couche controller : Gère les requêtes client et délègue le processus d'affichage aux vues qui sont dans la partie webapp de l'application. En quelques sorte un role de routeur.

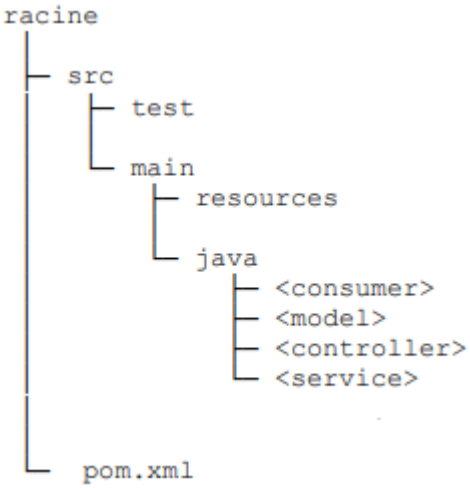
5.1.2 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

Les répertoires sources sont crées de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)



API Web



Web application

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

La gestion des logs est assurée à l'aide de Log4j.

6.2 - Fichiers de configuration

Le fichier application.yaml configure le webservice Spring. La base de données y est configurée. Par ailleurs le fichier dump_db.sql permet de charger, au lancement de l'API un jeu de données.

6.3 - Ressources

Les ressources sont stockées sur le serveur du client Web. Le client web accède à cet espace de stockage afin de charger ces différentes ressources.

6.4 - Environnement de développement

Ce Projet a été développé à partir des environnements Spring Boot 2.4.9, Angular et PostGreSQL 4.13 pour la base de données.

6.5 - Procedure de packaging / livraison

L'application sera packagée dans un fichier .war

Les scripts utilisés pour la création de la base de données seront livrés dans un fichier zip.

7 - GLOSSAIRE
