OC PIZZA

PIZZERIA OC

Dossier d'exploitation

Version 01

Auteur
Enyo TOVISSOU
Analyste Développeur

TABLE DES MATIÈRES

1 - VERSIONS

Auteur	Date	Description	Version
E. T	08/12/2019	Création du document	01

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application Pizzeria OC.

L'objectif de ce document est de fournir à l'entreprise ***OC PIZZA*** ou à son équipe technique, certaines informations techniques, essentielles pour le déploiement, la maintenance et l'utilisation de l'application ***PIZZERIA OC***.

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT - 01** : Dossier de conception technique de l'application
2. **DCF - 1.1** : Dossier de conception fonctionnelle de l'application.

3 - PRÉREQUIS

3.1 - Système

3.1.1 - Serveur de Base de données

Le SGBD utilisé par l'application Pizzeria oc est PostgreSQL dans sa version 12.1. La base de données sera hébergée sur des serveurs ***`Apache Tomcat.`***

3.1.1.1 - Caractéristiques techniques

Le fichier postgresql.jar doit être placé dans le répertoire `$CATALINA_HOME/lib` de Tomcat

3.1.2 - Serveur Web

Nous utiliserons comme serveur le conteneur d'application ***`Apache Tomcat.`***

3.1.2.1 - Caractéristiques techniques

`Apache Tomcat` ou **`serveur Tomcat`**, est un open-source conteneur de servlet java développé par **`Apache Software Foundation.`** Tomcat implémente plusieurs spécification de **JAVA EE**, y compris le **`Java Servlet`**, **`javaServer Pages, Java EL`** et **`WebSocket`**, et fournit une « pure java » http serveur web, environnement dans lequel java code peut exécuter.

3.2 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **Postgresql :** version 12.1
- **Avec son schéma de création :** create_db_oc_pizza.sql et son dump db_pizza_oc_dump.sql

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

L'application Pizzeria OC nécessite l'utilisation de **Google Maps Api** et d'un **système bancaire**.

Ces services doivent être à jour et accessibles pour le bon fonctionnement de l'application. Le premier permet la géolocalisation des filiales du groupe ainsi ses clients. Et le second permet de faire des transactions bancaires.

Pour une meilleure utilisation des services Tomcat, il est nécessaire d'installer Maven et de faire sa configuration notamment le chemin d'accès.

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de l'Application Web

4.1.1 - Artefacts

L'artefact de déploiement de l'application est fichier **`pizzeriaoc.war`** qui sera placé dans le répertoire webapps de notre serveur Tomcat.

4.1.2 - Environnement de l'application web

4.1.2.1 - Variables d'environnement

Le serveur d'application **`TOMCAT`** doit être exécuté avec la variable d'environnement suivante définie au démarrage. Elle est nécessaire afin de récupérer le répertoire contenant les fichiers de configuration de l'application :

-Dch.ocpizzeria.apps.conf=$CATALINA_HOME

INFO : il ne faut pas mettre de « / » à la fin de la valeur de la variable et ne pas utiliser d'espace dans le chemin.

4.1.3 - Répertoire de configuration applicatif

Le répertoire de configuration applicatif doit être créé sur le système de fichier et définit de la façon suivante :

$CATALINA_HOME/webapps/pizzeriaoc

- pizzeriaoc.war

4.1.4 - Data Sources

Les accès aux bases de données doivent se configurer à l'aide des fichiers de configuration de Tomcat.

Le fichier de drivers **postgresql (postgresql-12.1)** doit être déposé dans le répertoire :

$CATALINA_HOME/lib/

Dans le fichier context.xml de Tomcat, on doit configurer la ressource de la manière suivante :

```
<Context>
<Resource name="jdbc/postgres" auth="Container"
          type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
          url="jdbc:postgresql://127.0.0.1:5432/pizzeriaocdb"  username="myuser"
```

```
password="mypasswd" maxActive="20" maxIdle="10" maxWait="-1"/>
<Context>
```

Ensuite dans le web.xml de l'application ajouter ce qui suit :

```
<resource-ref>
 <description> postgreSQL Datasource in Pizzeria oc</description>
 <res-ref-name>jdbc/postgres</res-ref-name>
 <res-type>javax.sql.DataSource</res-type>
 <res-auth>Container</res-auth>
</resource-ref>
```

4.1.5 - Vérifications

```
InitialContext cxt =new InitialContext();
if (cxt == null ) {
   throw new Exception("Uh oh -- no context!");
}
DataSource ds = (DataSource) cxt.lookup( "java:/comp/env/jdbc/postgres" );
if ( ds == null ) {
   throw new Exception("Data source not found!");
}
```

Afin de vérifier le bon déploiement de l'application, faire ceci : **`mvn clean deploy`** depuis le répertoire $CATALINA_HOME/webapps/

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Base de données

La base de données démarre avec le démarrage de Tomcat : double clic sur startup.bat dans Tomcat/bin ou en line de commande taper startup.

5.2 - Application web

L'application est démarrée avec le démarrage du serveur.

Juste copier le fichier war (***pizzeriaoc.war***) dans $CATALINA_HOME/webapps/ , puis démarrer ou redémarrer tomcat.

6 - PROCÉDURE DE MISE À JOUR

6.1 - Base de données

La base de données peut être mise à jour depuis pgAdmin, l'interface de postgresql ou en exécutant un script depuis l'application en utilisant un sauvegarde ou un dump de la base de données. Un dump peut se faire depuis pgAdmin.

6.2 - Application web

L'application peut être mise à jour en déployant une mise à jour **`pizzeriaoc.war`** sur le serveur Tomcat.

Avec Maven, on peut depuis la ligne de commande : ***`mvn tomcat7 deploy`*** depuis le répertoire $CATALINA_HOME/webapps/

7 - SUPERVISION/MONITORING

7.1 - Supervision de l'application web

La surveillance des performances de Tomcat peut être effectuée soit en s'appuyant sur des Beans JMX, soit en utilisant un outil de surveillance dédié comme MoSKito ou JavaMelody.

Une façon d'obtenir les valeurs des MBeans est via l'application Manager fournie avec Tomcat. Cette application est protégée, donc pour y accéder, vous devez d'abord définir un utilisateur et un mot de passe en ajoutant les éléments suivants dans le fichier conf / tomcat-users.xml:

```
<role rolename="manager-gui"/>
<role rolename="manager-jmx"/>
<user username="tomcat" password="s3cret"roles="manager-gui, manager-jmx"/>
```

L'interface de gestion de l'application est accessible à l'adresse :

http://localhost:8080/manager/html.

Il contient des informations minimales sur l'état du serveur et des applications déployées, ainsi que la possibilité de déployer une nouvelle application, par exemple des fuites de mémoire dans toutes les applications déployées.

Des informations sur les beans JMX sont disponibles sur

`http://localhost:8080/manager/jmxproxy`

Ceci est affiché dans un format texte, car il est destiné à être traité par des outils.

Pour récupérer des données sur un bean spécifique, vous pouvez ajouter à l'URL des paramètres qui représentent le nom du bean et l'attribut souhaité :

`http://localhost:8080/manager/jmxproxy/?get=java.lang:type=Memory&att=HeapMemoryUsage`

7.2 - Supervision de la base de données

Nous avons vu au point 7.1 la supervision de du serveur et des applications qu'il héberge. Nous avons configuré notre base de données sur le même serveur Tomcat, et donc de cette façon, de la même manière nous que précédemment, nous pouvons superviser notre base données **`progresql`.**

Dans ce cas, on aura le bean, *`MBean Catalina:type=DataSource`*

8 - PROCÉDURE DE SAUVEGARDE ET DE RESTAURATION

8.1 - L'application web

Le déploiement se faisant avec un le fichier .war, et avec l'aide de maven, on a les commandes suivantes :

```
mvn tomcat7: deploy
mvn tomcat7: undeploy
mvn tomcat7:redeploy
```

8.2 - La base de donnée

La base de données peut être restaurée à partir d'un dump qui a été au préalable effectué. Un dump peut se faire sur l'interface de pgAdmin de postgresql, ou en ligne de commande ou dans un script de batch.

```
pg_dump pizzeriaoc_db > pizzeriaocdump.sql
```

ou

```
pg_dump pizzeriaoc_db -f pizzeriaocdump.sql
```

Et pour restaurer:

```
Createdb -Ttemplate0 pizzeriaoc_db
psql pizzeriaoc_db < pizzeriaocdump.sql
```

9 - GLOSSAIRE
