

## Manuel de déploiement

Ref doc : MDEP

Version : 1.0

Statut : approuvé

Ce document constitue le manuel de déploiement pour l'application GSA.

### Liste des participants au groupe projet

Nom	Initiales	Email	Appartenance	Qualité/Rôle
Ayoub El Yousfi	AE	ayoub.elyousfi@etu.univ-amu.fr	Université Aix-Marseille	Développeur
Joël Forward	JF	joel.forward@etu.univ-amu.fr	Université Aix-Marseille	Développeur
Mariana Andujar	MA	mariana.andujar@univamu.fr	IBDM	Responsable adjointe du service développement
Magali Contensin	MC	magali.contensin@univ-amu.fr	IBDM	Responsable du service développement
Pierre Vincent	PV	pierre.vincent.1@etu.univ-amu.fr	Université Aix-Marseille	Développeur
Mohamed Siraj Achabbak	SA	mohamed-siraj.achabbak@etu.univ-amu.fr	Université Aix-Marseille	Développeur
Youssef Jellab	YJ	youssef.jellab@etu.univ-amu.fr	Université Aix-Marseille	Développeur

### Liste de diffusion du document

Destinataire	Version(s) diffusée(s)	Date de diffusion de la dernière version
Participants	1.0	19/03/2019
<b>Restriction de diffusion</b>	Ce document ne doit pas être copié ou diffusé à un tiers hors de la liste de diffusion sans l'accord du chef de projet MOA	

### Historique des révisions du document

Version	Date révision	Page/sections concernées	Description de la modification	Auteur (initiales)	Date d'approbation	Approuvé par
1.0	19/03/2019	Toutes	Création	PV	19/03/2019	PV

## I. Prérequis

1. Un serveur Tomcat version 9.0.14 ou plus
2. Un serveur apache version 2.4.25 ou plus
3. MySQL Distrib 10.1.26-MariaDB ou plus
4. Maven
5. Angular-cli

## II. Configuration de l'application

Avant de compiler le projet il est nécessaire d'éditer le fichier de configuration 'application.properties'

Ce fichier se trouve ici : [GSA / gsa-server / src / main / resources / application.properties](#)

Il se présente comme ceci :

```
1  spring.jpa.hibernate.ddl-auto=create-drop
2  spring.datasource.url=jdbc:mysql://localhost/gsa?serverTimezone=UTC
3  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
4  spring.datasource.username=gsa
5  spring.datasource.password=gsa
6  #spring.jpa.show-sql=true
7  spring.datasource.initialization-mode=always
8  # here is the ip address of the computer in the tech area (local address if the app is on the same network)
9  # ----- app config -----
10 app.ipTechArea=http://localhost:4200
11 app.super-admin-password=tarkov
12 # ----- email -----
13 spring.mail.host=smtp.gmail.com
14 spring.mail.port=587
15 spring.mail.username=alerts.gsa@gmail.com
16 spring.mail.password=lespetitspedestres
17 spring.mail.properties.mail.smtp.auth=true
18 spring.mail.properties.mail.smtp.starttls.enable=true
19
```

### 1. Configuration de la base de données


Voici les informations nécessaires pour configurer l'application en fonction de votre base de données :

- La ligne 2 permet de configurer l'adresse d'accès à la base de données, par défaut celle-ci est sur localhost/gsa (la base de données se nomme 'gsa' par défaut)
- La ligne 4 et 5 permettent respectivement de renseigner le nom d'utilisateur et le mot de passe pour l'utilisateur de la base de données.

### 2. Configuration des variables de l'application

Les variables d'application servent à définir le comportement de l'application :

- La ligne 10 permet de définir l'adresse du poste défini comme étant le « local technique » (certaines fonctions de l'application ne sont disponible que depuis ce poste).
- La ligne 11 permet de définir le mot de passe superadministrateur (ce mot de passe sera requis pour passer l'application en mode maintenance)

	Projet	Document	Version	Date	p. 2 / 4
	GSA	MDEP	1.0	19/03/2019	

### 3. Configuration du serveur SMTP

Le serveur SMTP servira à envoyer les emails d'alertes, un compte google a été créé pour l'occasion mais il est recommandé d'utiliser un serveur interne à l'entreprise :

- La ligne 13 permet de définir l'adresse du serveur SMTP
- La ligne 14 permet de définir le port utilisé sur le serveur SMTP
- La ligne 15 permet de définir l'email utilisé avec le serveur SMTP
- La ligne 16 permet de définir le mot de passe pour le serveur SMTP

### 4. Configuration des contrôleurs

L'application étant un service REST, il faut autoriser les requête CORS provenant du serveur gérant les vues. Pour ce faire il faut ajouter l'adresse du serveur front-end aux autorisation CORS sur les contrôleurs 'AdminController' et 'UserController'.

Ces fichiers se trouvent ici :

[GSA](#) / [gsa-server](#) / [src](#) / [main](#) / [java](#) / [fr](#) / [uniamu](#) / [ibdm](#) / [gsa\\_server](#) / [controllers](#) /

Editez les fichiers 'AdminController.java' et 'UserController.java' en conséquence :

```
@CrossOrigin(allowCredentials = "true", origins = {"http://localhost:4200", "http://localhost", "http://51.77.147.140"})  
public class AdminController {
```

A l'intérieur de l'annotation '@CrossOrigin' ajoutez l'adresse du serveur apache dans la liste des origines et effacez les adresses inutiles.


### 5. Configuration du Front-End

Le font-end a besoin de connaître l'adresse du serveur REST. il faut donc le configurer dans le fichier 'environment.prod.ts' qui se trouve ici : [GSA](#) / [gsa-client](#) / [src](#) / [environments](#) / [environment.prod.ts](#)

Il se présente comme ceci :

```
1 export const environment = {  
2   production: true,  
3   API_URL: 'http://51.77.147.140:8080',  
4   API_ADMIN: '/admin'  
5 };  
6
```

La variable 'API\_URL' doit contenir l'adresse du serveur REST, éditez le fichier en conséquence.

	Projet	Document	Version	Date	p. 3 / 4
	GSA	MDEP	1.0	19/03/2019	

### III. Compilation et déploiement

#### 1. Serveur REST

La compilation du serveur REST (Spring-Boot) se fait via le gestionnaire de paquets Maven, la compilation produit un fichier .war qui peut être déployé sur un serveur Tomcat.

La compilation est réalisée en se plaçant dans le répertoire 'gsa-server' et en exécutant la commande suivante :

```
GSA/gsa-server$ mvn package
```

Cette commande va exécuter les tests unitaires et produire le fichier 'gsa\_server.war' dans le répertoire 'target'.

Il suffit ensuite de déployer le fichier .war dans le répertoire 'webapps' de votre serveur Tomcat.

Pour un environnement Linux :

```
3 mv gsa_server.war ROOT.war &&
4 cp ROOT.war <path_to_tomcat>/webapps/ &&
5 ./<path_to_tomcat>/bin/startup.sh
```

Pour un environnement Windows :

```
7 move gsa_server.war ROOT.war &&
8 copy ROOT.war <path_to_tomcat>/webapps/ &&
9 <path_to_tomcat>/bin/startup.bat
```

#### 2. Angular

La compilation du Front-End se fait via angular-cli grâce à la commande suivante : `GSA/gsa-client$ ng build --prod`

Une fois la commande exécutée, le contenu du dossier 'GSA/gsa-client/dist/gsa-client' doit être copié dans le dossier d'apache. Par défaut ce dossier se trouve être à l'emplacement '/var/www/html'.

Pour un environnement Linux : `11 cp -r dist/gsa-client/* /var/www/html/`


Pour un environnement Windows : `13 copy -r dist/gsa-client/* /var/www/html/`

Puisque le routage se fait via le routeur d'Angular Il faut également configurer apache pour rediriger toutes les requêtes vers index.html :

1. Créez un fichier '.htaccess' dans le répertoire '/var/www/' avec le contenu suivant :

```
16 RewriteEngine On
17 # If an existing asset or directory is requested go to it as it is
18 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
19 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d
20 RewriteRule ^ - [L]
21 # If the requested resource doesn't exist, use index.html
22 RewriteRule ^ /index.html
```

2. Activez le moteur de réécriture d'apache grâce à la commande suivante : `24 sudo a2enmod rewrite`
3. Relancez apache : `sudo service apache2 restart`

	Projet	Document	Version	Date	p. 4 / 4
	GSA	MDEP	1.0	19/03/2019	