|  |  |  |
| --- | --- | --- |
| Projet : **Gestion des Stocks d’Anticorps (GSA)** | | |
| **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 |
| **Restriction de diffusion** | 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 |

Table des matières

[I. Prérequis 3](#_Toc4363926)

[II. Configuration de l’application 3](#_Toc4363927)

[1. Configuration de la base de données 3](#_Toc4363928)

[2. Configuration des variables de l’application 4](#_Toc4363929)

[3. Configuration du serveur SMTP 4](#_Toc4363930)

[4. Configuration CORS 4](#_Toc4363931)

[5. Configuration du Front-End 4](#_Toc4363932)

[III. Compilation et déploiement 5](#_Toc4363933)

[1. Serveur REST 5](#_Toc4363934)

[2. Angular 5](#_Toc4363935)

# 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

# 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 :

Il se présente comme ceci :

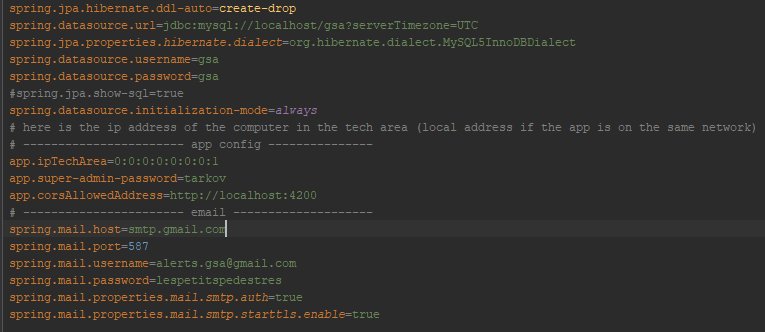


Figure 1 - application.properties

## Configuration de la base de données

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

* La variable spring.datasource.url 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)
* Les variables spring.datasource.username et spring.datasource.password permettent respectivement de renseigner le nom d’utilisateur et le mot de passe pour l’utilisateur de la base de données.

## Configuration des variables de l’application

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

* La variable app.ipTechArea 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 variable app.super-admin-password permet de définir le mot de passe superadministrateur (ce mot de passe sera requis pour passer l’application en mode maintenance)

## 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 variable spring.mail.host permet de définir l’adresse du serveur SMTP
* La variable spring.mail.port permet de définir le port utilisé sur le serveur SMTP
* La variable spring.mail.username permet de définir l’email utilisé avec le serveur SMTP
* La variable spring.mail.password permet de définir le mot de passe pour le serveur SMTP

## Configuration CORS

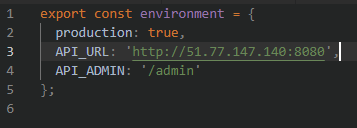
L’application étant un service REST, il faut autoriser les requête CORS provenant du serveur gérant les vues. Pour ce faire il faut ajouter l’adresse du serveur front-end aux autorisation CORS.

La variable app.corsAllowedAddress permet d’autoriser les requêtes CORS provenant de cette URL.

## Configuration du Front-End

Le font-end a besoin de connaitre l’adresse du serveur REST, il faut donc le configurer dans le fichier ‘environment.prod.ts’ qui se trouve ici :

Il se présente comme ceci :



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

# Compilation et déploiement

## 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 :



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 :



Pour un environnement Windows :



## Angular



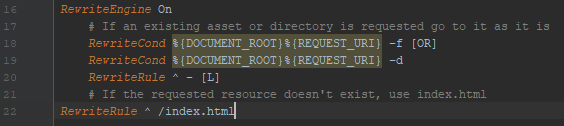
La compilation du Front-End se fait via angular-cli grâce à la commande suivante :

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 :

Pour un environnement Windows :

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 : 
2. Activez le moteur de réécriture d’apache grâce à la commande suivante :
3. Relancez apache : 