Compilation et déploiement

Compilation

```
cd /opt/esup-signature
# pour la première compilation il faut installer les dépendances de sedalib en lancant
mvn clean initialize
# puis par la suite mvn clean initialize ne sera plus utile, la commande suivante sera suffisante
mvn clean package
```

Vous devez lancer la commande **mvn clean package** pour obtenir dans ./target/ un fichier esup-signature.war (executable ou déployable) ainsi qu'un dossier esup-signature (déployable dans un tomcat)

Pour faire référence à un fichier de configuration situé en dehors du dépôt (peut être utile dans le cas de déploiements automatiques), il faut ajouter l'option -Dspring.config.location=/<DIR>/application.yml.

Lors de la compilation des tests d'intégration vont être exécutés, cela va vous permettre de contrôler votre configuration et votre environnement.

Si un test échoue (ERROR ou FAILURE) la compilation sera annulée.Dans le cas idéal tous les tests doivent passer.

Toutefois, selon votre configuration, certains tests sont évités (Skipped). Cela veut dire que l'application peut être déployée mais que certaines fonctionnalités seront inopérantes (envoi de mail, certains workflows...)

Voici un exemple de résultat obtenu :

```
[INFO] [INFO] Results: [INFO] [WARNING] Tests run: 11, Failures: 0, Errors: 0, Skipped: 4 [INFO] [INFO]
```

Dans ce cas les tests sont passés mais il y a des warnings. En remontant les logs on trouve par exemple :

```
2020-05-25 16:30:11.575 ERROR 3619 --- [me-limited test] o.e.esupsignature.WorkflowServiceTest : Test Workflow
: VisaAndSignWorkflowTest KO org.esupportail.esupsignature.exception.EsupSignatureUserException: ldap user not
found : user.test@univ-ville.fr
```

Il s'agit dans, cet exemple, d'une classe workflow (src/main/java/org/esupportail/esupsignature/service/workflow/impl/VisaAndSignWorkflowTest.java) qui est configurée avec un mail inconnu dans le LDAP. Une page dédiée explique le fonctionnement de ces classes paramétrable :

Créer une classe workflow

Il est possible d'éviter les tests en utilisant la commande **mvn clean package -DskipTests** mais des problèmes risquent de se poser lors du déploiement.

Déploiement / Lancement

Les deux premières méthodes sont à privilégier

Lancement de esup-signature.war

Depuis la version 1.16, il est possible de lancer directement le fichier war

```
./target/esup-signature.war
```

Dans ce cas c'est Spring qui démarre son propre tomcat embarqué avec les paramètres spécifiés dans le fichier de configuration (remote-ip-header, basedir, port, eventuellement ajp.port)

Il est possible d'externaliser le fichier de configuration à l'aide de l'option --spring.config.location=/<DIR>/application.yml

Paramétrage memoire :

```
export JAVA_OPTS=--add-exports=java.base/sun.security.x509=ALL-UNNAMED --add-exports=java.base/sun.security.
pkcs=ALL-UNNAMED -Xmx1024m -Xms1024
```

Spring boot

On peut aussi démarrer l'application directement avec la commande suivante au niveau du répertoire des sources. Dans ce cas c'est le fichier src/main /resources/application.yml qui sera pris en compte pour votre configuration.

De même que pour le lancement du war c'est spring qui lance son propre tomcat.

```
mvn spring-boot:run -Dspring.devtools.livereload.enabled=false
```

Pour spécifier un autre emplacement pour le fichier de configuration il faut ajouter -Dspring.config.location=/<DIR>/application.yml

Tomcat

> Attention, votre version tomcat doit correspondre à la version utilisée par spring boot. Depuis la version 1.27 l faut tomcat 10 minimum. Nous conseillons les deux premières méthodes de déploiement pour éviter cette gestion des versions.
>
> De plus, nous avons remarqué un problème (l'application plante lors des export SEDA) sur les instances installées dans un tomcat sous CentOs 7 (quel que soit la version de java et/ou tomcat). Le problème ne se reproduit pas sous Debian ou en utilisant le tomcat embaqué dans Spring Boot

Pré-requis : Tomcat version 10 minimum pour esup-signature 1.27, Tomcat 9 pour les versions précédentes.

Copier soit le fichier target/esup-signature.war vers webapps/ROOT.war du tomcat ou directement le contenu du dossier target/esup-signature/ vers webapps/ROOT/ obtenu après la compilation

```
rm -rf /opt/tomcat-esup-signature/webapps/ROOT && cp -rf /opt/esup-signature/target/esup-signature /opt/tomcat-
esup-signature/webapps/ROOT
```

On arrête le tomcat avant et on le redémarre ensuite

```
/opt/tomcat-esup-signature/bin/shutdown.sh
```

Paramétrage mémoire JVM :

Pensez à paramétrer les espaces mémoire JVM :

export JAVA_OPTS="-Xms2048M -Xmx2048M -Djava.awt.headless=true --add-exports=java.base/sun.security.x509=ALL-UNNAMED --add-exports=java.base/sun.security.pkcs=ALL-UNNAMED"

Pour maven :

export MAVEN_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m"

Démarrage :

```
/opt/tomcat-esup-signature/bin/startup.sh
```

Bravo, l'installation est terminée !

Pour aller plus loin, RDV sur la page dédiée à l'exploitation : https://www.esup-portail.org/wiki/display/SIGN/Exploitation