Manuel d'installation du Front Office esup-smsu

- Installation du Front office
  - Paramétrage Base de donnée et Hibernate
  - Paramétrage Web services
  - Paramétrage LDAP
  - Paramétrage adhésion
  - Paramétrage destinataires
  - Paramétrage envoi de SMS
  - Paramétrage authentification
  - Paramétrage SMTP
  - Paramétrage des purges
  - Paramétrage des traitements asynchrones
  - Test de connexion back office
- Création de la base de donnée
  - Création du schéma en base de donnée
- Soit lancement simple avec jetty
- Soit déploiement de l'application
- Configuration de la langue des mails de modération

Installation du Front office

L'application est disponible sur github. Il est conseillé d'utiliser git pour télécharger les sources :

```
git clone https://github.com/EsupPortail/esup-smsu.git
```

N'hésitez pas également à utiliser GIT en interne pour exploiter et maintenir à jour vos instances.

Vous pouvez aussi trouver des zip sur cette page : https://github.com/EsupPortail/esup-smsu/tags.

Paramétrage des propriétés de l'application

Le front office se déploie en mode servlet ou portlet.

Créez src/main/resources/properties/config.properties en s'inspirant de config.sample.properties. si vous utilisez git pour l'exploitation et la mise à jour, il est conseillé de faire :

```
ln -s config.sample.properties src/main/resources/properties/config.properties
git add src/main/resources/properties/config.properties
git commit -m 'utiliser config.sample.properties comme base de configuration'
```

Vérifier la configuration des fichiers :

- src/main/resources/properties/config.properties
  - vérifier les informations de connexion à la base de donnée
  - vérifier les informations de connexion au LDAP
  - vérifier l'URL d'accès au web service du portail
  - vérifier l'URL d'accès au web service du back office
  - vérifier le chemin d'accès au fichier libmgs.properties
  - vérifiez les informations de configuration du compte de validation
  - dans le cas d'un déploiement en servlet, vérifiez le paramétrage du serveur CAS
  - vérifiez le paramétrage SMTP
  - vérifiez le paramétrage Quartz
  - de manière générale, vérifiez l'ensemble des paramètres.
- src/main/resources/properties/logging/log4j.properties
  - Vérifier le chemin d'accès au fichier de log

Voici le détail des propriétés :

Paramétrage Base de donnée et Hibernate

- Url, login et mot de passe d'accès à la base de donnée du front office :

```
hibernate.connection.jdbc.url=jdbc:mysql://<host>:3306/smsu
hibernate.connection.jdbc.username=root
hibernate.connection.jdbc.password=xxxx
```

Paramétrage Web services

- Adresse web service back office :

```
smsuapi.ws.address=https://[Nom_De_La_Machine_Back_Office]:[Port_WebService]/smsu-api/
```

(attention, le "/" à la fin est important)

Paramétrage LDAP

- Adresse, login, mot de passe et timeout (en millisecondes) du serveur LDAP :

```
ldap.url=ldap://nomServeurLdap:389
ldap.userName=cn=sms,ou=admin,dc=univ-paris1,dc=fr
ldap.password=xxx
ldap.connectTimeout=5000
```

- La base DN du serveur LDAP :

```
ldap.base=dc=univ-paris1,dc=fr
```

- Le DN sub path pour les utilisateurs :

```
ldap.dnSubPath=ou=people
```

- L'identifiant d'utilisateur dans le LDAP :

```
ldap.uidAttribute=uid
```

- Le nom de l'attribut qui caractérise le nom « d'affichage de l'utilisateur » (généralement son nom et prénom concaténé dans un attribut LDAP) :

```
ldap.displayNameAttribute=displayName
```

- Le nom de l'attribut qui contient le prénom de l'utilisateur :

```
ldap.firstNameAttribute=givenName
```

- Le nom de l'attribut qui contient le nom de famille de l'utilisateur :

```
ldap.lastNameAttribute=sn
```

- Le nom de l'attribut qui contient l'adresse email de l'utilisateur :

```
ldap.emailAttribute=mail
```

- Le nom de l'attribut qui contient le numéro de téléphone portable de l'utilisateur :

```
ldap.pagerAttribute=pager
```

- Le nom de l'attribut qui contient la liste des conditions d'utilisation qui ont été acceptées par l'utilisateur :

```
ldap.termsOfUseAttribute=up1TermsOfUse
```

- La valeur qui préfixera les valeurs stockées dans le LDAP à l'attribut défini par ldap.termsOfUseAttribute :

```
ldap.termsOfUseAttributeEtiquetteSMSU={SMSU}
```

Ce préfixe à la mode SUPANN permet de partager l'attribut défini par ldap.termsOfUseAttribute avec d'autres applications. Le service SMS-U gardera inchangé les valeurs ne commençant pas par ce préfixe.

- La valeur qui sera stockée dans le LDAP à l'attribut défini par ldap.termsOfUseAttribute lorsque l'utilisateur accepte les conditions générales :

```
ldap.key.cg=CG
```

NB : cette valeur sera préfixée de la valeur défini par ldap.termsOfUseAttributeEtiquetteSMSU

- Le nom de l'attribut sur lequel sont effectuées les recherches d'utilisateurs par nom :

```
ldap.searchAttribute=cn
```

- L'expression permettant d'effectuer des tests :

```
ldap.testFilter=cn=*aaron*
```

Ce paramètre défini l'expression qui sera utilisé pour tester le LDAP par le biais des taches ant livrées par esup-commons

- Le nom de l'object dont se sert le serveur LDAP pour définir un utilisateur :

```
ldap.objectClass=Person
```

- Le DN sub path permettant de caractériser les groupes :

```
ldap.group.dnSubPath=ou=groups
```

- Le nom de l'attribut utilisé comme identifiant pour les groupes :

```
ldap.group.idAttribute=cn
```

- Le nom de l'attribut qui contient la liste des membres d'un groupe :

```
ldap.group.groupMemberAttr=member
```

- Le nom de l'attribut utilisé lors des recherches par nom sur les groupes :

```
ldap.group.groupSearchAttr=description
```

- Le nom de l'attribut affiché lors des recherches par nom sur les groupes :

```
ldap.group.groupSearchDisplayedAttr=description
```

- Le nom de l'attribut utilisé pour affiché le nom d'un groupe :

```
ldap.group.nameAttr=description
```

NB : supprimé dans esup-smsu 1.1.2

- Le nom de l'object dont se sert le serveur LDAP pour définir un groupe :

```
ldap.group.groupObjectClass=groupOfNames
```

- L'expression permettant d'effectuer des tests sur les groupes :

```
ldap.group.testFilter=cn=*mati*
```

Ce paramètre défini l'expression qui sera utilisé pour tester le LDAP par le biais des taches ant livrées par esup-commons

Paramétrage adhésion

- Activation/Désactivation de la validation par SMS du numéro de téléphone d'un adhérent :

```
adhesion.activateValidation=false
```

- Nombre maximum utilisé dans la génération des codes de validation :

```
adhesion.maxNumberCodeValidation=100000
```

- Compte d'imputation des messages de validation :

```
adhesion.accountValidation=[compte_de_validation]
```

- Rôle associé au compte de validation (non utilisé, présent pour la cohérence des données en base :

```
adhesion.roleValidation=[role_compte_validation]
```

- Titre du SMS de validation (corps du message envoyé) :

```
adhesion.titleSmsValidation=[Titre_SMS_valdidation]
```

Paramétrage destinataires

- Expression régulière de validation de la forme du numéro de téléphone d'un adhérent :

```
recipient.phoneNumberPattern=0[67]\\d{8}|\\+33\\s[6|7](\\s\\d\\d){4}
```

(10 chiffres commençant par 06.
Ce paramètre peut être vidé (pas de vérification du tout)

Paramétrage envoi de SMS

- Compte par défaut d'envoi de SMS :

```
sms.defaultAccount=defaut
```

- Login du superviseur par défaut :

```
sms.defaultSupervisorLogin=admin
```

Paramétrage authentification

- URL serveur CAS :

```
cas.url=https://[Nom_De_La_Machine_Cas]:[Port_Cas]/cas
```

- nom de la machine et port pour accéder au front-office :

```
server.url=http://localhost:8080
```

Paramétrage SMTP

- L'adresse du serveur SMTP à utiliser :

```
smtp.host=an.smtp.host
```

Ce paramètre défini l'adresse du serveur SMTP qui sera utilisé par le front office pour envoyer les emails
- Le port du serveur SMTP à utiliser :

```
smtp.port=25
```

Ce paramètre défini le port du serveur SMTP qui sera utilisé par le front office pour envoyer les emails
- Le login a utiliser pour s'authentifier auprès du serveur SMTP :

```
smtp.user=
```

Ce paramètre défini le login a utiliser pour s'authentifier auprès du serveur SMTP qui sera utilisé par le front office pour envoyer les emails
- Le mot de passe a utiliser pour s'authentifier auprès du serveur SMTP :

```
smtp.password=
```

Ce paramètre défini le mot de passe a utiliser pour s'authentifier auprès du serveur SMTP qui sera utilisé par le front office pour envoyer les emails
- L'adresse email qui sera utilisée pour envoyer les emails:

```
smtp.fromEmail=example@domain.org
```

Ce paramètre défini l'adresse email qui sera utiliser pour envoyer les emails
- Le nom qui apparaitra dans les emails :

```
smtp.fromName=Name firstName
```

Ce paramètre défini le nom qui apparaitra comme expéditeur des emails envoyés par le front office.

Paramétrage des purges

- Purge de la base des utilisateurs en attente de validation de numéro de téléphone :

```
purge.pendingMember.seniorityDay=30
```

- purge de la base des messages :

```
purge.periodic.seniorityDay=120
```

Paramétrage des traitements asynchrones

- La cron expression utilisée par la tache de supervision d'envoi des sms :

```
quartz.superviseSmsSendingTrigger.cronExpression= 0 0 * * * ?
```

NB : cette tache cron n'est plus vraiment utile.
Ce paramètre défini l'expression cron qui est utilisée pour planifier la tache qui supervise l'envoi des SMS au back office.

- La cron expression utilisée par la tache de purge des utilisateurs en attente de validation de numéro de téléphone :

```
quartz.purgePendingMemberTrigger.cronExpression=0 0 3 1 *  ?
```

- La cron expression utilisée par la tache de purge des messages :

```
quartz.periodicPurgeTrigger.cronExpression=0 0 3 2 * ?
```

- La cron expression utilisée par la tache d'envoi de mail aux utilisateurs dont le numéro de téléphone est en erreur :

```
quartz.notificationByMailForInvalidPhoneTrigger.cronExpression=0 0 12 ? * MON
```

Test de connexion back office

- Test de connexion avec le back office. En cas de succès, le nom de l'application cliente lue dans le certificat est retourné :

```
welcome.isConnexionTested=false
```

Création de la base de donnée

Le back office nécessite un serveur de base de donnée MySQL en version 5.

Création du schéma en base de donnée

Pour créer le schéma de base de donnée : se connecter au serveur mysql en tant qu'administrateur et saisir le mot de passe

```
mysql -u root -p -e "create database smsu"
mysql -u root -p smsu < src/main/resources/database/create_tables.sql
mysql -u root -p smsu < src/main/resources/database/create_qrtz_tables_esup-smsu.sql
```

Nb : Aucune table ne doit être présente dans le schéma smsuapi au moment de l'exécution de cette commande sous peine d'échec.

Initialisation des tables en base de données

Il faut premièrement configurer le compte par défaut ainsi que le premier super administrateur. Pour ce faire, éditer le fichier «**src/main/resources/database/populate_tables_esup-smsu.sql**», puis modifier les lignes suivantes

```
INSERT INTO account VALUES (1, "default account");
```

Il faut remplacer la valeur "default account" par le nom du compte par défaut (défini dans le fichier de configuration config.properties par la clef ***smsuapi.defaultAccount***

```
INSERT INTO customized_group VALUES (1, 1, 1, "admin", 1, 1, 1);
```

Il faut remplacer la valeur "admin" par le login de la personne qui sera administrateur.

Puis faire :

```
mysql -u root -p smsu < src/main/resources/database/populate_tables_esup-smsu.sql
```

Vérification des tables en base de données

Pour vérifier que les étapes précédentes se sont correctement déroulées : se connecter à la base de donnée et saisir le mot de passe :

```
mysql -u root -p -e "show tables" smsu
```

La liste des tables doit apparaître de la manière suivante :

```
+--------------------------+
| Tables_in_smsu           |
+--------------------------+
| QRTZ_BLOB_TRIGGERS       |
| QRTZ_CALENDARS           |
| QRTZ_CRON_TRIGGERS       |
| QRTZ_FIRED_TRIGGERS      |
| QRTZ_JOB_DETAILS         |
| QRTZ_JOB_LISTENERS       |
| QRTZ_LOCKS               |
| QRTZ_PAUSED_TRIGGER_GRPS |
| QRTZ_SCHEDULER_STATE     |
| QRTZ_SIMPLE_TRIGGERS     |
| QRTZ_TRIGGERS            |
| QRTZ_TRIGGER_LISTENERS   |
| account                  |
| b_vers_mana              |
| basic_group              |
| customized_group         |
| fonction                 |
| mail                     |
| mail_recipient           |
| message                  |
| pending_member           |
| person                   |
| recipient                |
| role                     |
| role_composition         |
| service                  |
| supervisor               |
| supervisor_sender        |
| template                 |
| to_mail_recipient        |
| to_recipient             |
+--------------------------+
```

Soit lancement simple avec jetty

```
mvn jetty:run
# ou ant jetty.run
```

Soit déploiement de l'application

```
mvn package
```

puis copiez le war dans webapps (servlet).

ou si vous préférez vous pouvez utiliser ant en configurant préalablement le chemin de déploiement dans build.properties

```
ant deploy
```

Ajouter l'application dans le contexte du serveur du portail, par exemple par le biais du fichier conf/Catalina/localhost/esup-smsu.xml

```
<Context docBase="/usr/local/esup-smsuapi/tomcat/webapps/esup-smsu" >
  <Resource
   name="jdbc/esup-smsu"
   auth="Container" type="javax.sql.DataSource" driverClassName="com.mysql.jdbc.Driver"
   url="jdbc:mysql://localhost/smsu" username="xxx" password="xxx"
   maxActive="100" maxWait="10000"
   validationQuery="select 1"
   removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />
</Context>
```

(le contexte peut aussi être configuré dans le fichier tomcat conf/server.xml)

Configuration de la langue des mails de modération

Les mails de modération sont traduits, mais la langue utilisée dépend de la locale système. Pour avoir ces messages en français, il faut s'assurer de lancer tomcat/jetty avec la locale système configurée, par exemple en mettant LANG=fr_FR.UTF-8 .