Manual d'installation du Back Office (esup-smsu-api)

Installation du Back office

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

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

Paramétrage

Le back office se déploie en mode servlet.
Configurez les fichiers :

- 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'
      ```

    - pour les paramètres, cf **paramètres généraux** ci-dessous
- src/main/resources/properties/logging/log4j.properties
    - vérifier le chemin d'accès du fichier de log
- src/main/resources/properties/broker/broker.xml
    - et aussi src/main/resources/properties/broker/olm/libmgs.properties si vous utilisez le broker OLM

Paramètres généraux

Voici la liste des paramètres disponibles :

- Url d'accès à la base de donnée :

  ```
  hibernate.connection.jdbc.url=jdbc:mysql://<host>/smsuapi
  ```

  Ce paramètre définit l'url de la base de donnée du back office.
- Login d'accès à la base de donnée :

  ```
  hibernate.connection.jdbc.username=root
  ```

  Ce paramètre définit le login pour l'accès à la base de donnée du back office.

- Mot de passe d'accès à la base de donnée :

```
hibernate.connection.jdbc.password=xxx
```

Ce paramètre définit le mot de passe pour l'accès à la base de donnée du back office.

- Mode d'accès à la base de donnée :

```
hibernate.useJndi=false
```

Défini le mode d'accès à la base de donnée du back office.

- Nom du connecteur utilisé :

```
sms.connector.name=allmysms
```

Ce paramètre définit le nom du broker utilisé. Les brokers disponibles sont "olm", "smsenvoi", "proxy" et "allmysms".

- Active / désactive l'envoi effectif des SMS :

```
sms.connector.simulateSending=false
```

Ce paramètre sert à désactiver l'envoi effectif des messages au broker. Si à « true » alors aucun message ne sera envoyé au broker.

- Nombre maximum de jour de conservation des SMS :

```
purge.sms.seniorityDay=180
```

Ce paramètre sert à définir la durée maximum (en jours) de conservation des SMS en base avant que ceux ci ne soient purgés.

- Nombre maximum de jour de conservation des statistiques :

```
purge.statistic.seniorityDay=730
```

Ce paramètre sert à définir la durée maximum (en jours) de conservation des statistiques en base avant que ceux ci ne soient purgés. Ces statistiques sont utilisés pour la création des rapports consolidés.

- La cron expression utilisée par la tache de génération des statistiques :

```
quartz.buildStatisticsTrigger.cronExpression=0 0 0 1  * ?
```

Ce paramètre définit l'expression cron qui est utilisée pour planifier la tache qui génère les statistiques nécessaires aux relevés consolidés.

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

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

Ce paramètre définit l'expression cron qui est utilisée pour planifier la tache qui purge les SMS.

Paramètres liés au broker

Les brokers disponibles sont "olm", "smsenvoi", "proxy" et "allmysms".

Broker allmysms

Voici la liste des paramètres disponibles dans le fichier src/main/resources/properties/config.properties :

- Utilisateur et mot de passe de connexion à allmysms.com (fourni par allmysms.com) :

```
sms.connector.allmysms.account.login=xxx
sms.connector.allmysms.account.apikey=xxx
```

- Pour configurer le champ expéditeur d'un SMS. Attention, ce champ est limité : 11 caractères max et caractères alphanumériques+espace

```
sms.connector.allmysms.from.mapJSON = { "": "Univ Xxxxxx", "compte1": "Foo" }
```

Broker OLM

Voici la liste des paramètres disponibles dans le fichier src/main/resources/properties/broker/olm/libmgs.properties :

- Identifiant de l'application (fourni par OLM) :

```
libmgs.smsuapi.sid=1234
```

- Chemin d'accès du certificat utilisé pour communiquer avec OLM (fourni par OLM) :

```
libmgs.smsuapi.https.keystore=/an/example/certificat.ks
```

- Mot de passe du certificat (fourni par OLM) :

```
libmgs.smsuapi.https.passwd=xxxxxxx
```

- Type de logger :

```
libmgs.smsuapi.log.type=file
```

Défini le type de logger utilisé par la librairie.

- Fichier de log :

```
libmgs.smsuapi.log.file.path=/an/example/pushLibMGS.log
```

Défini le chemin d'accès du fichier de log utilisé par la librarie libMGS.

- Timeout de notification :

```
libmgs.smsuapi.advanced.notifTimeout=1
```

Défini la durée (en minute) avant que la connexion vers l'url de notification soit automatiquement fermée et ré-ouverte.

Broker smsenvoi

Voici la liste des paramètres disponibles dans le fichier src/main/resources/properties/config.properties :

- Utilisateur et mot de passe de connexion à smsenvoi.com (fourni par smsenvoi.com) :

```
sms.connector.smsenvoi.account.email = smsu@univ-paris1.fr
sms.connector.smsenvoi.account.apikey = xxx
```

- Pour configurer le champ expéditeur d'un SMS. Attention, ce champ est limité : 10 caractères max et caractères alphanumériques+espace

```
sms.connector.smsenvoi.from.mapJSON = { "": "Univ Xxxxxx", "compte1": "Foo" }
```

- Intervalle en secondes de connexion à smsenvoi.com pour obtenir les statuts

```
sms.connector.smsenvoi.acknoledgeStatus.repeatInterval=20
```

Pour obtenir le statut d'un SMS (En cours, Délivré, Numéro invalide...), le back office doit se connecter à smsenvoi.com à intervalle régulier.

Broker "proxy"

- URL de back office SMS-U proxy

```
sms.connector.proxy.ws.address=https://sms.univ-xxx.fr/
```

- Utilisateur et mot de passe de connexion à smsenvoi.com (fourni par smsenvoi.com) :

```
sms.connector.proxy.ws.username = xxx
sms.connector.proxy.ws.password = xxx
```

- Intervalle en secondes de connection au proxy pour obtenir les statuts

```
sms.connector.proxy.acknoledgeStatus.repeatInterval=20
```

  Pour obtenir le statut d'un SMS (En cours, Délivré, Numéro invalide...), le back office doit se connecter au back office "proxy" à intervalle régulier.

Création de la base de données

Le back office nécessite un serveur de base de données 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 smsuapi"
mysql -u root -p smsuapi < src/main/resources/database/create_tables.sql
mysql -u root -p smsuapi < src/main/resources/database/create_qrtz_tables_esup-smsuapi.sql
```

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

Vérification des tables en base de données

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

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

La liste des tables doit apparaître de la manière suivante :

```
+--------------------------+
| Tables_in_smsuapi        |
+--------------------------+
| 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                  |
| application              |
| blacklist                |
| institution              |
| sms                      |
| statistic                |
+--------------------------+
```

Déploiement

```
mvn package
```

puis copiez le war dans webapps.

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-smsuapi.xml

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

Remarque : <Resource> n'est pas nécessaire si on utilise hibernate.useJndi=false

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

L'application peut être déployée dans un serveur d'applications existant, voir la documentation du framework esup-commons

Lancement de l'application

en test

L'application se lance par la tache

```
ant jetty.run

# or
mvn jetty:run
```

en production

L'application se lance quand on lance le serveur tomcat.

Pour envoyer plus de 10000 SMS, il faut configurer le paramètre tomcat "maxParameterCount".

Les services REST d'esup-smsu-api sont synchrones : esup-smu-api ne répond que lorsque l'opération demandée est effectivement traitée. Aussi si une requête sur smsu-api correspond à demander à envoyer un sms à plusieurs milliers de numéros de téléphones, le temps de réponse peut être long. Il faut donc s'assurer qu'aucun timeout au niveau d'un éventuel reverse-proxy ne vienne perturber le bon renvoi de la réponse par esup-smus-api.

Sous apache, on positionnera un **timeout sur le proxypass suffisamment conséquent** : par exemple 1 heure et non 1 minute comme positionné par défaut.

```
ProxyPass / ajp://localhost:8009/ retry=1 timeout=3600
```

Configuration sans esup-smsu-api-admin

La façon recommandée d'administrer esup-smsu-api est d'utiliser esup-smsu-api-admin.

Il est néanmoins possible de s'en passer, en modifiant directement la base de données smsu-api. Pour faire la configuration initiale à la main, vous pouvez faire :

```
insert into institution (INS_LABEL) values ("monUniv");
insert into account (ACC_LABEL, ACC_QUOTA) values ("test", 100);
insert into application (INS_ID, ACC_ID, APP_NAME, APP_CERTIFCATE, APP_QUOTA) values (1, 1, "test", "xxx", 20);
```

Problèmes

Erreur de connexion au broker OLM

Si vous rencontrez le warning suivant dans smsuapi.log :

```
Warning sent by lib SGS : - arg0 : Problem while sending data.(javax.net.ssl.SSLHandshakeException : Received
fatal alert: handshake_failure)
```

ajoutez

```
-Dsun.security.ssl.allowUnsafeRenegotiation=true
```

au lancement de tomcat.

NB : problème rencontré sur java 1.6.12