

Configurations

Dans src/main/resources/security.properties :

Depuis la maj du 09/05/2022 il faut définir les adresses ip autorisées pour accéder aux WS Rest :
(séparées par des virgules)
- adresse(s) d'esup-nfc-tag-server
- adresse(s) autorisée(s) à accéder aux ws json comme par exemple https://esup-emargement.univ-vile.fr.fr/wsrest/export/session/..

Dans src/main/resources/esup-emargement.properties :

BASE DE DONNÉES

Esup-emargement utilise une base de données pour stocker les données de l'application.

La base créée, la construction de la table peut se faire en démarrant une première fois esup-emargement avec dans src/main/resources/esup-emargement.properties le paramètre **spring.jpa.hibernate.ddl-auto** à **create**.

Une fois ce premier démarrage effectué et donc la table créée, remettez **spring.jpa.hibernate.ddl-auto** à **update** afin que la base ne soit pas écrasée à chaque redémarrage.

```
emargement.datasource.driver-class-name=org.postgresql.Driver
emargement.datasource.jdbc-url=jdbc\:postgresql\://localhost\:5432/emargement
emargement.datasource.username=esupemargement
emargement.datasource.password=esup
```

Une base **postgresql** est nécessaire - ne serait-ce que par rapport aux requêtes utilisées pour réaliser les graphiques / courbes de statistiques ; par défaut ces requêtes sont propres à postgresql, cf ci-dessous.

Graphiques / courbes de statistiques

Attention, les graphiques (courbes) des statistiques sont construits grâce à des requêtes SQL qui sont liées au SGBD : nous n'avons pas pu développer de requêtes SQL indépendant du SGBD utilisé.

CAS

Configuration du serveur CAS pour l'authentification.

```
cas.key=CAS_PROVIDER_LOCALHOST_9000
cas.url=https://cas.univ-ville.fr
cas.service=https://esup-emargement.univ-ville.fr
```

A noter qu'il faut explicitement dire à CAS d'exposer certains des attributs LDAP auxquels il accède (cf la doc de cas sur "Attribute Release Policies"), notamment "eduPersonPrincipalName"

Il faut absolument, pour les performances, indexer eduPersonPrincipalName (eppn) dans le ldap et la base de données.

LDAP

Configurations Ldap pour :

- vérifier que l'utilisateur en base existe bien lors d'un ajout
- récupération du d'attributs comme nom, prénom
- recherche prédictive utilisateurs
- recherche de groupes pour import d'individus



```
spring.ldap.urls=ldap://ldap.univ-ville.fr
spring.ldap.username=cn=xxxxx,dc=univ-ville,dc=fr
spring.ldap.password=
spring.ldap.base=dc=univ-ville,dc=fr
ldap.groups=ou=groups

#Filtre permettant la récupération de l'utilisateur ldap en fonction de l'uid renvoyé par CAS
ldap.userFilter=(uid=%s)
```



Dans le fichier **src/main/java/org/esupportail/emargement/domain/userLdap.java** :

```
@Entry(base = "ou=people", objectClasses = {"inetOrgPerson" })
```

On ne peut pas ici utiliser le fichier properties pour valuer @Entry dans Spring Ldap ODM:

<https://github.com/spring-projects/spring-ldap/issues/444>

Egalement, les infos récupérées dans le Ldap sont configurées dans cette même classe :

```
private @Attribute(name = "sn") String name;
private @Attribute(name = "mail") String email;
private @Attribute(name = "eduPersonPrincipalName") String eppn;
private @Attribute(name = "cn") String nomPrenom;
private @Attribute(name = "displayName") String prenomNom;
private @Attribute(name = "givenName") String prenom;
private @Attribute(name = "supannEtuld") String numEtudiant;
private @Attribute(name = "supannCivillite") String civilite;
private @Attribute(name = "supannEmpId") String numPersonnel;
```

Ici on peut changer la valeur de l'attribut correspondant au champ demandé.

Ex : ici on récupère le N° d'étudiant à partir du supannEtuld, si c'est à partir d'un autre attribut, faire le changement.

SUPERADMINS



Point d'entrée de l'application elle correspond à un filtre LDAP récupérant une liste de personne,

Exemples :

```
((uid=admin)(uid=esup))
```

```
(memberOf=cn=xxxxx,ou=groups,dc=univ-ville,dc=fr)
```

```
#Superadmins --> filtre ldap
ldap.superAdminFilter=( |(uid=admin)(uid=esup))
```

APOGEE

Apogée est utilisé lors de la recherche de groupes d'individus à importer dans l'application.

```
emargement.datasource.apogee.jdbc-url=jdbc:oracle:thin:@apogee.univ-ville.fr:1522/APOGEE
emargement.datasource.apogee.username=
emargement.datasource.apogee.password=
emargement.datasource.apogee.driver-class-name=oracle.jdbc.OracleDriver
```

Les requêtes pour la récupération des données sont dans la classe **src/main/java/org/esupportail/emargement/services/ApogeeService.java**

SMTP

Configuration pour l'envoi de mail : convocations, consignes, logs.

```
spring.mail.host=smtp.univ-ville.fr
spring.mail.port=25
spring.mail.username=
spring.mail.password=
spring.mail.properties.mail.smtp.auth=false
spring.mail.properties.mail.smtp.starttls.enable=false
```

PHOTO

La photo est affichée lors de l'émargement en lui même.

```
emargement.wsrest.photo.prefixe = https://esup-sgc.univ-ville.fr/wsrest/photo/
emargement.wsrest.photo.suffixe = /photo
```

LOGS

Dans le fichier **src/main/resources/logback.xml** : chemin vers le répertoire de logs, envoi de mail possible...

Cron de purge des logs ainsi que l'ancienneté des logs en jours, à effacer.

```
log.cron.purge = 0 0 10 * * * // Appelé tous les jours à 10H
log.all.retention = 90 //nombre de jours de rétention
```

APPLICATION

L'url de l'application est utilisé dans la partie Calendar.

```
app.url = https://esup-emargement.univ-ville.fr
app.nomDomaine = univ-ville.fr
```

ESUP-NFC-TAG

Liens vers les clients esup-nfc-tag. ils seront affichés dans la page Administrateur/applications

```
emargement.esupnfctag.link.jar = https://esup-nfc-tag.univ-ville.fr/nfc-index/download-jar
emargement.esupnfctag.link.exe = https://esup-sgc.univ-ville.fr/esup-sgc-client-installer.zip
```

WEB SERVICE

Le fichier **security.properties** permet la restriction de l'accès aux WS pour l'accès à esup-nfc-tag et aux API d'export. C'est une liste d'adresses IP séparées par des virgules.

```
accessRestrictionWSRest= 127.0.0.1
```

ADE Campus

Si l'option Ade Campus est activée dans les configurations, vous pourrez créer des sessions à partir d'évènements.

Si l'option *emargement.ade.api.url.encrypted* est renseignée, cette url sera utilisée à la place de *emargement.ade.api.url* pour la connexion à la webapi.

L'url "cryptée" est à récupérer depuis les paramètres d'Ade Campus.

```
emargement.ade.home.url = https://ade.univ-ville.fr
emargement.ade.api.url.encrypted =
emargement.ade.api.url = https://ade.univ-ville.fr/jsp/webapi
emargement.ade.api.login =
emargement.ade.api.password =
```

Esup-signature

Si l'option (ESUPSIGNATURE_ENABLED) esup-signature est activée dans les configurations, vous pourrez envoyer un Pdf d'émargement dans esup-signature afin qu'il soit signé par les personnes autorisées.

Celles-ci sont désignées également dans les configurations (ESUPSIGNATURE_EMAILS).

L'id correspond à celui du circuit de signature

```
emargement.esupsignature.url = https://esup-signature.univ-ville.fr
emargement.esupsignature.workflow.id = 1250
```