Documentation utilisateur Ynov DaP

# Mise en place du serveur

Le serveur doit être lancé via la commande « java -jar billemaz\_maxime\_dap\_server.jar »

Pour fonctionner, un sous-dossier « dap » doit être présent dans le dossier user.home (variable en fonction du système d’exploitation) et il doit contenir un fichier credentials.json (fourni par google api). Un dossier tokens sera créé et contiendra le fichier de crédentials auto-généré par l’application.

De même, un fichier application.properties peut être placé dans un dossier « config » à côté du jar. Ce fichier contient les informations de connexion à la base de données ainsi que quelques propriétés optionnelles :

* - config.appName : Nom de l’application (Défaut DaP)
* - config.root=Dossier (chemin absolu) où seront placés les dossiers suivants (Par défaut le dossier dap dans user.home)
* - config.credentials = Dossier contenant le fichier credentials.json
* config.token=tokens = Dossier contenant le fichier de token

Par défaut, la configuration pour la base de données est la suivante :

spring.datasource.url=jdbc:mysql://localhost:3306/dap

spring.datasource.username=DapUser

spring.datasource.password=DapPassword

spring.datasource.driverClassName = com.mysql.jdbc.Driver

spring.jpa.database=MYSQL

spring.jpa.show-sql=true

spring.jpa.generate-ddl=true

useSSL=false

verifyServerCertificate=false

Un exemple pour ce fichier est fourni à la racine de bim.server.

Le serveur se lancera sur un TomCat sur le port 8080

Voici la liste des chemins utilisables dans l’application :

* /user/add/{userKey} : ajoute l’utilisateur {userKey}
* /account/add/google/{accountName}?userKey={user}: ajoute un compte google a l’utilisateur {user}
* /account/add/Microsoft/{accountName}?userKey={user}: ajoute un compte Microsoft a l’utilisateur {user}
* /email/nbUnread ?userKey={user} : affiche le nombre de mails non lu dans tous les comptes de l’utilisateur
* /email/google/nbUnread ?userKey={user} : affiche le nombre de mails non lu dans tous les comptes google de l’utilisateur
* /email/Microsoft/nbUnread ?userKey={user} : affiche le nombre de mails non lu dans tous les comptes Microsoft de l’utilisateur
* /contact/number ?userKey={user} : affiche le nombre de contacts dans tous les comptes de l’utilisateur
* /contact/google/number ?userKey={user} : affiche le nombre de contacts dans tous les comptes google de l’utilisateur
* /contact/Microsoft/number ?userKey={user} : affiche le nombre de contacts dans tous les comptes Microsoft de l’utilisateur
* /event/next ?userKey={user} : affiche le prochain evenement dans tous les comptes de l’utilisateur
* / event/google/next ?userKey={user} : affiche le prochain évènement dans tous les comptes google de l’utilisateur
* / event/Microsoft/next ?userKey={user} : affiche le prochain évènement dans tous les comptes Microsoft de l’utilisateur

Deux URL’s sont dotés d’une interface utilisateur :

/admin : Affiche tous les comptes google et Microsoft stockés en base ainsi que leurs informations

/email/Microsoft/nextMessages?userKey={user} : affiche les 10 prochains mails de tous les comptes Microsoft de l’utilisateur.

# Utilisation du client

De la même manière, le client doit se lancer via la commande « java -jar billemaz\_maxime\_dap\_client.jar »

Lancer le client via cette commande sans arguments affichera un message d’aide indiquant les différents paramètres possibles (Ce message apparaitra aussi si des paramètres incorrects sont rentrés).

Deux paramètres sont obligatoires pour utiliser correctement le client :

* L’action à effectuer. Actuellement, quatre actions sont possibles :
  + addUser : Ajoute un nouvel utilisateur à notre application qui sera autorisé à envoyer des requêtes via l’API. Si une des autres actions est utilisée via un utilisateur non crée, la requête ne pourra pas s’effectuer.
  + addGoogleAccount : Ajoute un nouveau compte google a l’utilisateur spécifié
  + addMicrosoftAccount : Ajoute un nouveau compte Microsoft à l’utilisateur spécifié
  + getNbUnreadMail : affiche le nombre de mails non lu dans la boite principale.
  + getNextEvent : affiche le nom, la date de début, la date de fin et le statut du prochain évènement du calendrier google.
  + getNbContact : affiche le nombre de contacts du compte google.
* Le nom de l’utilisateur (normalement crée via un addUser)
* Le troisième paramètre est optionnel, il permet d’effectuer les requêtes sur un autre compte google que celui qui est actuellement connecté.