ZANOTTI LOIC

DEVOIR NUMERO 2 JAVA

**CLIENT :**

Une première commande d’utilisation du jar permet d’avoir un exemple d’utilisation, pour ceci la commande est :

* java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar ”help”

**Utilisation du client :**

* API et ajout d’un utilisateur : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar [options] [userKey]
* Usage ajout d’un compte google ou Microsoft sur notre utilisateur :
  + Java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar [options] [userKey] [accountName]

**List des options disponible :**

* API
  + **« mail-unread »** Affiche les mails des compte microsoft + google non-lus

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « mail-unread » « loic »

* + **« events »** Affiche le prochain évènement prévu dans mes agendas google + microsoft

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « events » « loic »

* + **« contacts »** Affiche le nombre de contacts de mes comptes google + microsoft

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « « contacts » « loic »

* + **« view »** Affiche les résultats des 3 options précédentes
  + Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « view » « loic »
* USER :
  + « add-user » Ajoute un utilisateur en bdd sur lequel nous pourrons assigner des comptes google et microsoft

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « add-user » « myUserKey »

* COMPTE GOOGLE :
  + « add-google-account » Ajoute un compte google à l’utilisateur de crée via la route « add-user »

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « add-google-account » « myUserKey » « myAccountName »

* COMPTE MICROSOFT :
  + « add-microsoft-account » Ajoute un compte google à l’utilisateur de crée via la route « add-user »

Exemple : java -jar .\ zal.client.cmdline-0.0.1-SNAPSHOT.jar « add-microsoft-account » « myUserKey » « myAccountName »

SERVER:

**FICHIER CREDENTIALS.JSON :**

* Par défaut, le fichier credentials.json doit être placé dans le répertoire : C:\dap\google\credentials
* Si vous souhaitez modifier l’emplacement du répertoire par défaut :
  + Aller dans le fichier « src/main/resources/spring.properties »
  + Modifier la ligne « credentials\_file\_path »

**FICHIER STOREDCREDENTIAL :**

Le fichier « StoredCredential » généré lors de la connexion à un compte google est aussi externalisé.

* Par défaut, le fichier StoredCredential s’enregistrera dans le répertoire :

C:\dap\google\tokens

* Si vous souhaitez modifier l’emplacement du répertoire par défaut :
  + Aller dans le fichier « src/main/resources/spring.properties »
  + Modifier la ligne « credentials\_token\_path »

**BASE DE DONNEES :**

* La configuration des paramètres de base de données se fait dans le fichier « src/main/resources/application.properties »
* La configuration est basé par défaut sur un MySQL sur le port 3306 avec un utilisateur root sans mot de passe. La base de données est nommée « dap »

**INFORMATIONS APPLICATION MICROSOFT :**

* La configuration d’application microsoft se fait dans le fichier « src/main/resources/application.properties »
* Renseignez les champs correspondant aux credentials que microsoft vous a fourni lors de l’inscription de l’application.

**INTERFACE ADMINISTRATEUR :**

* Sur l’application vous trouverez un interface administrateur sur la route « localhost :8080/admin ». Au travers de cet interface vous aurez accès en visuel aux différentes données d’utilisateur comme :
  + Credentials Google et microsoft
  + Mails de vos compte microsoft enregistrés
  + Contacts de vos comptes microsoft

ATTENTION : pour toutes les routes /admin/[route]?userKey=

VOUS DEVREZ METTRE LA USERKEY SOUHAITE APRES LE « = »

**ROUTES SERVEUR API :**

Api google :

* « localhost:8080/google/mail/unread?userKey=myUserKey » Affiche les emails non-lus de vos compte google
* « localhost:8080/google/events/nextEvent?userKey=myUserKey » Affiche le prochain événement prévu de vos compte google
* « localhost:8080/google/contacts/number?userKey=myUserKey » Affiche le nombre de contacts de vos compte google

Api microsoft :

* « localhost:8080/outlook/mail/unread ?userKey=myUserKey » Affiche les emails non-lus de vos compte microsoft
* « localhost:8080/outlook/events/nextEvent?userKey=myUserKey » Affiche le prochain événement prévu de vos compte microsoft
* « localhost:8080/outlook/contacts/number?userKey=myUserKey » Affiche le nombre de contacts de vos compte microsoft

Ajout Compte Utilisateur :

* « localhost :8080/add/user?userKey=myUserKey » Ajoute un utilisateur avec la userKey (ici ce sera myUserKey) »

Ajout des comptes Google et Microsoft :

* « localhost :8080/add/account/google/accountName?userKey=myUserKey » cette route ajoute un compte google à votre utilisateur en bdd , ne pas oublier de préciser un accountName qui correspond au nom de votre compte et la userKey qui correspond à votre compte utilisateur de l’application enregistré en bdd.
* « localhost :8080/add/account/microsoft/accountName?userKey=myUserKey » cette route ajoute un compte microsoft à votre utilisateur en bdd , ne pas oublier de préciser un accountName qui correspond au nom de votre compte et la userKey qui correspond à votre compte utilisateur de l’application enregistré en bdd.

API Microsoft et Google :

* « localhost :8080/contacts?userKey=myUserKey » Affiche le nombre de contacts de vos compte enregistrées dans l’application (microsoft et google)
* « localhost :8080/mails/unread?userKey=myUserKey » Affiche le nombre d’emails non-lus de vos compte enregistrées dans l’application (microsoft et google)
* « localhost :8080/events/nextEvent?userKey=myUserKey » Affiche le prochain événement de vos compte enregistrées dans l’application (microsoft et google)