Guide d’utilisateur

# Généralité

L’application « DaP » permet de se connecter à l’API de Google pour récupérer différentes informations.

Elle est composée de deux parties :

* Serveur DaP : gère toutes les requêtes API Google et l’authentification
* Client DaP : se connecte au serveur ci-dessus et traite le résultat pour être lisible par l’utilisateur.

# Besoin système

L’application (serveur et cliente) est compatible avec tous les systèmes d’exploitation : Linux, Windows et MacOS.

Pour utiliser l’application vous aurez besoin sur votre ordinateur de :

* Java 1.8
* MySQL (avec une base de données)
* Un navigateur internet
* Un compte Google
* Un compte Microsoft
* Une console
* Un accès internet
* Le port 8080 disponible sur votre ordinateur

# Utilisation

## Lancement du serveur

Avant tout, vous avez besoin d’une base de données, par défaut le serveur cherche une base MySQL en local sur le port 3306, avec comme configuration :

* User : « dap »
* Mot de passe : « dap2018 »
* Base de données : « dap »
* L’utilisateur « dap » doit avoir pleins privilèges (FULL) sur la base de données « dap » en lecture, écriture, modification des données et structures.

**Configuration « application.properties » :**

Ce fichier est entièrement optionnel ! Ça marchera par défaut avec les paramètres ci-dessous. Sinon il faut placer le fichier « application.properties » à côté du « .jar » serveur, ou dans le dossier « config » à côté du « .jar ».

Valeurs modifiables (utilisées) :

|  |  |  |
| --- | --- | --- |
| **Clé** | **Défaut** | **Description** |
| spring.datasource.url | jdbc:mysql://localhost:3306/dap | URL base de données |
| spring.datasource.username | dap | User base de données |
| spring.datasource.password | dap2018 | Mot de passe base de données |
| spring.datasource.driverClassName | com.mysql.jdbc.Driver | Driver SQL |
| spring.jpa.database | MYSQL | Type de base |
| spring.jpa.generate-ddl | true | Générer les tables |
| spring.jpa.show-sql | true | Afficher les requêtes |
| spring.datasource.continue-on-error | true | Continuer si une erreur se produit |
| useSSL | false | Utiliser le SSL pour la base de données |
| verifyServerCertificate | false | Vérifier la signature de l’hôte de la base |
| config.configDir | DOSSIER HOME USER ACTIF | Dossier racine qui contient le credentials.json et auth.properties |
| config.credentialsFileName | credentials.json | Nom du fichier de crédentials de Google |
| config.authPropertiesName | auth.properties | Nom du fichier de configuration de l’application Microsoft |
| config.applicationName | JeanHeude | Nom de notre application |

**Pour lancer le serveur, vous devez :**

* Télécharger le JAR du serveur (but.server.jar) sur votre ordinateur
* Créer un dossier « dap » dans votre dossier « home » de l’utilisateur connecté
* Dans le dossier « dap » mettre VOTRE fichier « credentials.json » (à télécharger sur le Google Console API)
* Dans le dossier « dap » mettre VOTRE fichier « auth.properties » qui contient :

appId=MICROSOFT\_APP\_ID

appPassword=MICROSOFT\_APP\_PASSWORD

redirectUrl=http://localhost:8080/authorize

* + Remplacer les valeurs par les vôtres
* Ouvrir une console dans le dossier qui contient le JAR ci-dessus
* Lancer la commande « java -jar but.server.jar »
* Vous devriez voir « Started Launcher in X seconds » si tout c’est bien passé

Par défaut, il cherchera le « credentials.json » dans le dossier « dap » de votre home, et il sauvegardera les tokens dans le dossier « dap/token »

Le serveur écoute ensuite sur le port 8080 en localhost. Ce port ne doit pas être utilisé par une autre application.

## Utilisation du client

Pour utiliser les fonctions du client vous devez :

* Télécharger le JAR du client (buathier\_thibault\_dap\_client.jar) sur votre ordinateur
* Ouvrir une console dans le dossier qui contient le JAR ci-dessus
* Lancer la commande « java -jar but.client.jar »
* Une aide devrait être affichée

Le client peut prendre plusieurs arguments, des options et une commande sous le format suivant : « java -jar buathier\_thibault\_dap\_client.jar [args] <commande> ».

Pour plus d’aides : lancer dans le terminal « java -jar buathier\_thibault\_dap\_client.jar -help »

##### **Pour utiliser l’application, vous devez créer un compte DAP :**

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER addUser »

##### **Puis ajouter des comptes Microsoft et/ou Google :**

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER -account=MON\_NOM\_DE\_COMPTE addGoogleAccount »

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER -account=MON\_NOM\_DE\_COMPTE addMicrosoftAccount »

#### Afficher le prochain événement

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER showNextEvent »

Par défaut le calendrier « primary » sera utilisé, vous pouvez spécifier un calendrier avec l’argument « -calendar=CALENDRIER\_ID »

#### Afficher le nombre de mail non lu dans la boite principale

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER showEmailUnreadCount»

#### Afficher le nombre de contact

« java -jar buathier\_thibault\_dap\_client.jar -user=MON\_USER showContactCount»

## **URL du site « à ouvrir dans un navigateur » :**

<http://localhost:8080/admin> 🡺 Administration des utilisateurs et leurs comptes

[http://localhost:8080/microsoft/mail?userKey=MON\_USER](http://localhost:8080/microsoft/mail?) 🡺 Afficher ses mails (UNIQUEMENT via les comptes microsoft)