# Java – Maven

# Exercice 1 : Gestionnaire de mails

# Énoncé :

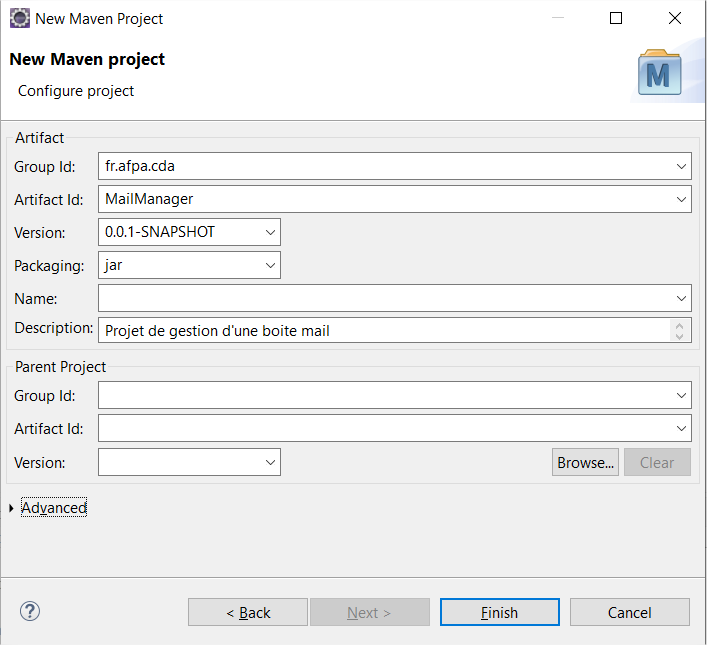
1 - Créer un projet Maven simple nommé « MailManager » qui sera packagé sous forme de jar avec les informations suivantes :

Group Id : fr.afpa.cda

Artifact Id : MailManager

Une image contenant texte

Description générée automatiquement



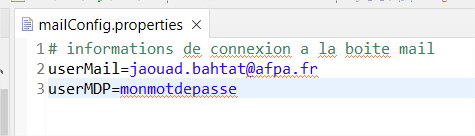
2- Créer un package « fr.afpa.cda.metier » dans les dossier sr/main/java et src/test/java

3- Rechercher sur internet les informations et ajouter les dépendances suivantes dans le pom.xml dans leur dernière version stable

* Junit (avec le scope « test »
* slf4j

Vérifier que les dépendances ont bien été téléchargées dans le dossier « Maven Dependencies »

4- Créer un fichier de propriétés dans le dossier src/main/resources nommé « mailConfig.properties » et y mettre les propriétés « user » et « mdp » (avec des valeurs de tests)



5- Créer une classe « ConnexionManager » avec une méthode nommée « validationMDP » qui prend en paramètre un String représentant un mot de passe et dont le rôle est de vérifier qu’il contient au minimum 8 caractères.

Si c’est le cas elle retourne un boolean vrai sinon false

6- Créer une classe de test Junit (dans le bon emplacement) pour tester la méthode « validationMDP » avec des assertions vérifiant les 2 cas (mot de passe valide et mot de passe invalide) à l’aide des méthode « assertTrue » et « assertFalse »

Exemple : assertTrue(validationMDP(‘’motdepasse’’) );

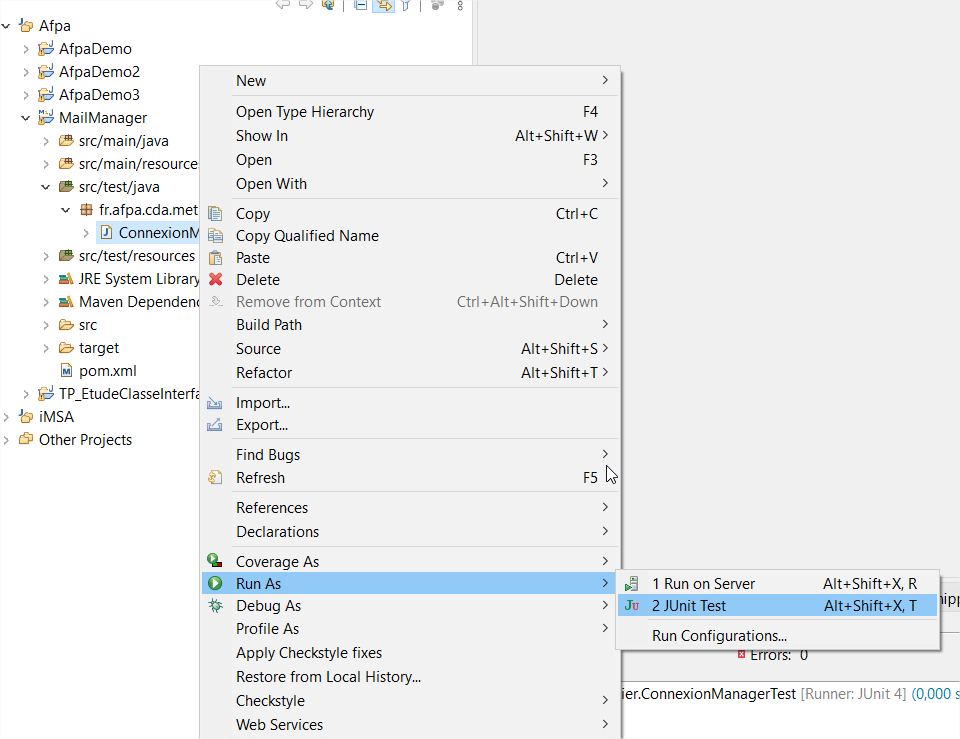
Puis exécuter le test unitaire

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement



7- Créer une autre méthode « connexionMail » qui récupère les informations du fichier properties et fait appel à « validationMDP » avec le mot de passe récupéré, si celle-ci retourne false déclencher une exception interne préalablement créée nommée « InvalidMDPException » avec le message « Le mot de passe est invalide »

8- Créer une autre classe nommée « ConnexionLauncher » qui contient une méthode main qui fait appel à la méthode « connexionMail » et qui en cas d’exception la capture pour afficher le message d’erreur qu’elle contient dans un fichier de log

Placez le fichier log4j.properties fourni sans le dossier src/main/resources

Exemple d’utilisation du logger :

**private** **static** **final** Logger logger = LoggerFactory.getLogger(ConnexionLauncher.**class**);

Puis dans la méthode :

} **catch** (**final** Exception e) {

LOGGER.error(e.getMessage(), e);

}