Rapport du Projet

Implémentation d'un client POP3

DIALLO Mohamed

2017-2018

1. Introduction

Protocole: **POP3**

1.1 Interface utilisateur

L'interface utilisateur est entièrement en mode console.
Elle s'inspire du client par défaut de Linux, fourni avec une invite de commande. Les actions sont accessibles en tapant le mot clé "help" après le lancement.
Les messages de sortie sont redirigés vers la sortie standard, et les erreurs vers la sortie d'erreur du système.

2. Usage

Pour tester le fonctionnement du protocole, il est nécessaire de lancer le serveur correspondant au client**.**

2.1 Serveur

Lancement du serveur POP3

1. aller dans le dossier server/
2. lancer l'exécutable PopServer.class : java PopServer, ou lancer le makefile associé.

Après la fermeture de la connexion client, il est nécessaire de relancer le serveur.

2.2 Client

Lancement du client POP3

1. aller dans le dossier class/
2. lancer l'exécutable pop3.jar : java -jar pop3.jar, ou lancer le makefile associé.

Un message de bienvenue doit s'afficher, avec en début d'invite de commande le texte pop3 >

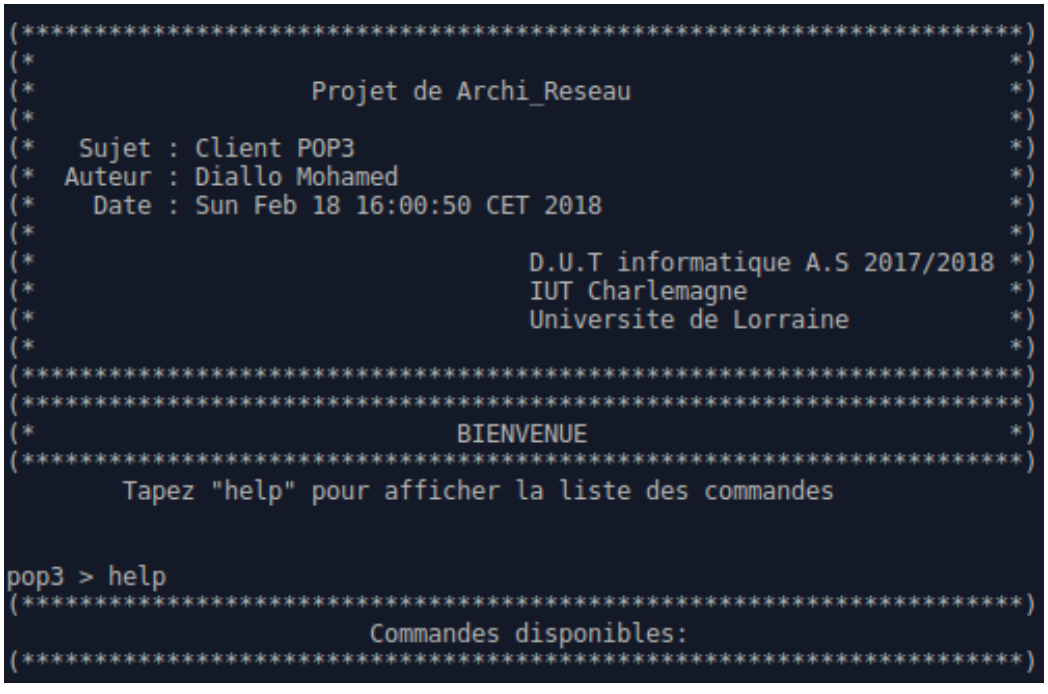

Figure 1 : Invite de commande pour le client POP3

3. Structure logique
3.1 POP3

Le diagramme de classe suivant présente la structure logique du client POP3.

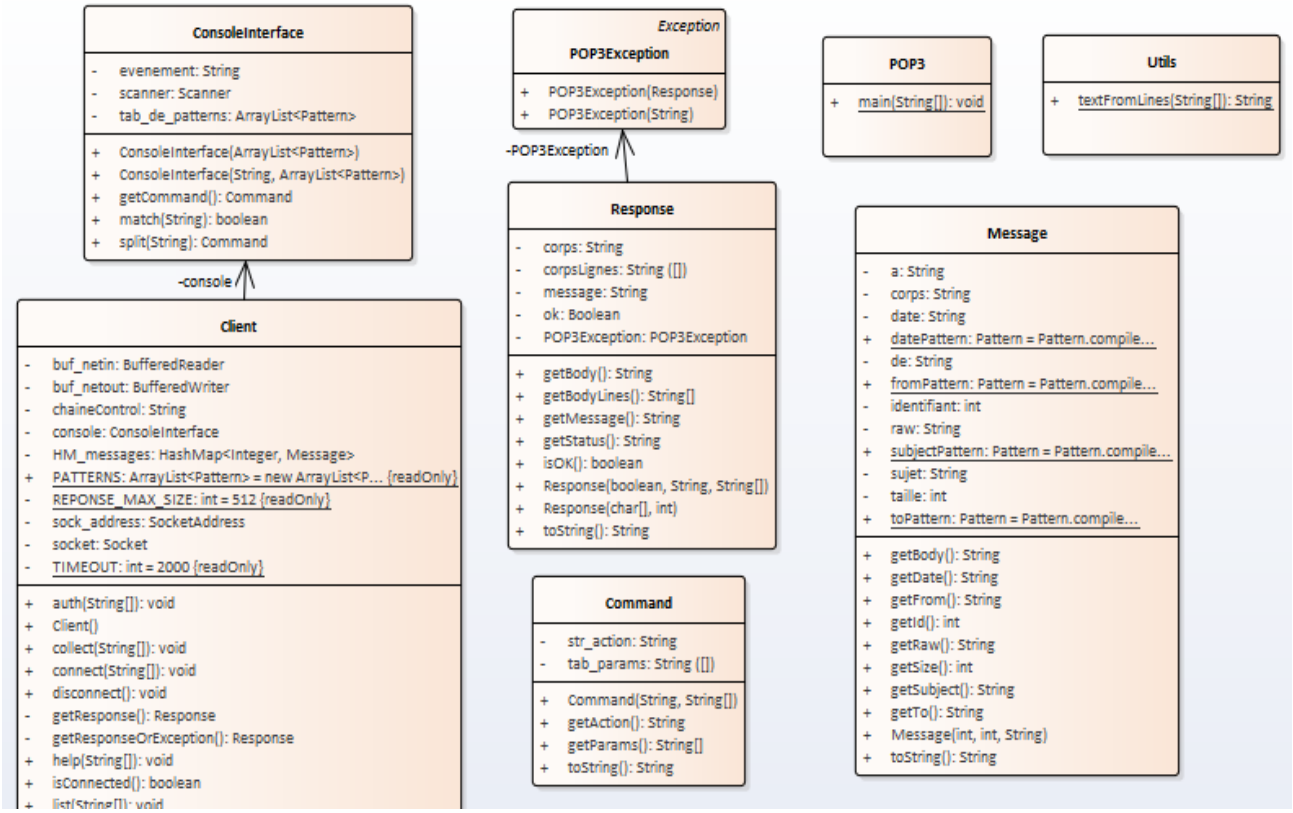

Figure 2 : Diagramme de classe du client POP3

Le point d'entré est la classe **POP3**. Il crée la classe client et appelle la méthode *run*.
La classe **Client** dispose de toutes les commandes POP3 supportées (*connect*, *auth* …). Ces méthodes spéciales prennent un tableau de paramètre en entrée et sont appelées grâce à un patron de conception (méthode *runCommand*) et (méthode *run*).

Les méthodes *sendCommand* et *getResponse* servent à gérer la réception et l'envoi de messages au serveur. Ils sont codés dans par les classes **Message** et **Response**.
En cas d'erreur, on lève une instance de la classe **POP3Exception**.

4. Structure physique

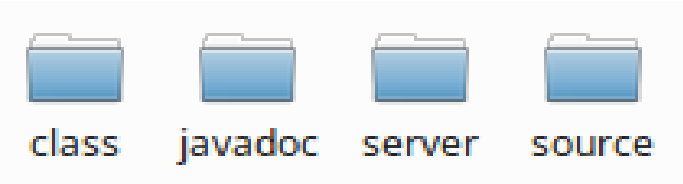

Figure 3: Dossiers du projet

1. class: exécutable Java
2. javadoc: documentation du projet (javadoc)
3. server: exécutable Java du serveur (format .class)
4. source: source du projet (.java)

5. Difficultées

L'exécution dépend de la réception et de l'envoi de paquets, ainsi que du système d'exploitation. Cependant, la difficulté que j'ai rencontrée est la mise en place de l'authentification "APOP", vu qu'on n'a pas assez parlé de son fonctionnement en cours (algorithme MD5) et la RFC n'est a priori pas très illustratif sur ce point. Donc, j'ai essayé, malgré tout, vous pouvez consulter ce que j'ai pu faire dessus, dans le fichier "client java", la méthode est en commentaire.

6. Remarque

J'ai remarqué aussi que le protocole APOP n'est pas implémenté sur le fichier serveur "PopServer" qui est fourni, parce que lors des tests, il affichait:

```
+OK bonjour alice
PASS 123456
+OK bienvenue alice !
APOP
-ERR Sorry, command APOP not implemented.
```

En conclusion, le projet était simple (sauf apop), mais j'ai essayé de le faire soigneusement et de lire attentivement les RFC pour comprendre les subtilités du protocole POP3.