**Rapport**

* Partie exécution de la commande :

Au début j’ai cherché à créer un objet de type Jsch qui va me servir ensuite dans l’exécution des commandes shell ou aux scripts sur le serveur. Après la création de cet objet j’ai essayé d’ouvrir une nouvelle session avec le nom d’utilisateur ainsi que l’hôte et le port. Puis j’ai défini le mot de passe et j’ai fait l’appel à connecter.

L’appel s’est fait via la commande session.connect() qui a pour rôle d’ouvrir une nouvelle connexion au serveur SSH distant. Une fois la connexion établie, vous pouvez lancer une nouvelle chaîne.

Après je vais essayer de créer un canal d’exécution qu’on va lui attribuer un InputStream.

On peut maintenant définir la commande qu’on veut exécuter. On peut aussi lire la sortie du flux d’entrée via la commande BufferedReader reader. Si on desire lire chaque ligne du lecteur tamponné et ajoutez-la à la liste des résultats on peut aussi le faire.

J’ai aussi utilisé intexitStatus afin de récupérer le statut de sortie de la commande à distance.

L’utilisateur peut également se déconnecter en toute sécurité du canal et déconnecter la session.

* Partie web Socket :

Comme on sait déjà que j’ai besoin de créer un canal full-duplex sur un socket TCP pour les navigateurs et les serveurs web, j’ai eu recours au protocole WEBSOCKET qui va m’aider à le développer. Grace à ce Protocol je serai capable d’ouvrir une connexion permanente entre le navigateur et le serveur et du coup ce dernier peut envoyer des requêtes au navigateur pour lui prévenir s’il y a des nouveaux.

Pour répondre à mes besoins j’ai créé 3 méthodes : OnMessage, OnOpen, OnClose.

* OnOpen elle sera chargée de nous informer la présence d’un client connecté.
* OnClose me permettra de fermer la connexion entre le client et le serveur.
* OnMessage se chargera de faire l’appel du script à exécuter et après son exécution elle renvoie un message au client pour enfin appeler la méthode OnClose pour fermer la connexion.
* Partie client :

En ce qui concerne la partie client dés qu’il soit présent une instance de la classe Websocket sera créée et la méthode OnOpen sera déclenchée d’une manière systématique ainsi phénomène expliqué précédemment se produira et un canal full duplex sera crée entre le serveur et le client. Le serveur restera en état d’écoute au cas de recevoir un message de la part du client si un message lui arrive la méthode OnMessage côté serveur sera déclenchée et elle va fonctionner de la manière expliquée précédemment.

La méthode OnMessage côté client va recevoir la réponse du serveur et se charge de l’afficher sur la page web.

NB : Il faut remplacer ces 3 attributs par des attributs de votre part lors du test.

**private static String USERNAME = "makdoudi";**

**private static String PASSWORD = "kom";**

**private static String HOST = "localhost";**