

TP n°8

Protocole SMTP en JAVA

Introduction

SMTP pour “SIMPLE MAIL TRANSFER PROTOCOL” définit les règles à suivre pour qu’un Email soit transmis, d’un ordinateur client (qui veut envoyer un Email) à un serveur (qui se charge de envoyer le mail au destinataire). L’envoi d’Email est basé sur le principe suivant : un client SMTP fait une requête d’expédition d’Email que le serveur SMTP accepte. Ce serveur peut être la destination finale ou peut n’être qu’un intermédiaire. Les commandes SMTP sont générées par le client SMTP et sont destinées au serveur SMTP. A chaque étape le serveur envoie une réponse pour signaler si la commande a été exécutée correctement.

La procédure classique d’envoi d’Email :

1. Client SMTP : Envoi de la commande MAIL indiquant qui envoie l’Email (remarquer qu’il est tout à fait possible de vous faire passer pour un autre).
2. Serveur SMTP : Si le serveur peut accepter des Email, il répond OK.
3. Client SMTP : Envoi de la commande RCPT indiquant à qui l’Email est destiné.
4. Serveur SMTP : Si le serveur accepte un Email avec cette adresse de destinataire, il répond encore OK. Sinon, il répond avec un code de rejet mais la transaction n’est pas rejetée au complet, simplement ce destinataire.
5. Client SMTP : Éventuellement, le client envoie d’autres adresses Email.
6. Client SMTP : Envoi de la commande DATA pour indiquer que le client va envoyer le contenu du mail. Pour marquer la fin de l’Email, le client enverra une séquence spéciale de caractères (. tout seul sur une ligne).
7. Serveur SMTP : Renvoie un OK s’il a bien reçu l’Email qu’il doit envoyer au destinataire.

Les commandes de bases :

- Commande HELO : est utilisée pour ouvrir la connexion entre le client et le serveur et d’autre part, pour identifier la machine client auprès du serveur SMTP.

Syntaxe : « HELO *hostname* » où *hostname* est le nom de la machine client (l’hôte à partir duquel vous vous connectez)

- Commande **MAIL** : indique au serveur de Mail qu’une nouvelle transaction viens de commencer. Cette information permet d’identifier l’expéditeur de l’Email. Si le serveur accepte l’adresse de l’expéditeur, il renvoie un OK avec le code 250.
Syntaxe : « **MAIL FROM:** *expéditeur* » où *expéditeur* est l’adresse Email de l’auteur du message.
- Commande **RCPT** : pour indiquer un destinataire du Email. Si le destinataire est accepté par le serveur, il renvoie le code 250. Il est possible d’avoir plusieurs destinataires pour un même Email, pour cela, il suffit de répéter cette commande autant de fois que nécessaire.
Syntaxe : « **RCPT TO:** *destinataire* » où *destinataire* est l’adresse Email du destinataire du message.
- Commande **DATA** : indique que toutes les lignes qui suivront jusqu’à la séquence de caractères qui mettra fin au corps du message (le caractère «.» tout seul sur une ligne) feront partie du message Email à envoyer. Le serveur renvoie le code 354 comme réponse(s) intermédiaire(s), ceci correspond à un OK ; un deuxième OK (avec le code 250) sera donné après la séquence de fin du corps du message, ceci signifie que le serveur à bien reçu le message Email qu’il doit envoyer au(x) destinataire(s). Le message Email peut commencer par zero, un ou plusieurs champs d’en-tête de la forme : « **Date:** », « **Subject:** », « **To:** », « **Cc:** », « **From:** ».
Syntaxe : « **DATA** »
- Commande **QUIT** : Permet de clôturer la session courante et de fermer les voies de communication entre le client et le serveur. Cette commande renvoie un OK avec le code 221.
Syntaxe : « **QUIT** »

Remarque : Vous avez déjà rencontré le protocole SMTP au cours du TP1.

Exercice 1 [Client SMTP]

Écrivez un programme Java `jmailer` qui

1. se connecte au service smtp du serveur `lucien`,
2. envoie un courrier électronique
3. lit et affiche les réponses du serveur (une réponse d’une ligne par “ordre”)

N’oubliez pas de traiter les codes de retour attendus et d’afficher un message d’erreur s’ils ne correspondent pas au bon comportement.