Marouane Abakarim

Mohamed Adansar

**TP 1 : Conception d'Applciation Réparties (CAR) Serveur FTP**

1. **Introduction :**

Le TP de Serveur FTP l'objectif et de la réalisation d'un serveur FTP avc lanquage JAVA qui devait fonctionner avec un client FTP (command FTP,FileZilla ,ect ..).

il faut avoir un serveur qui fait l'ecoute sur la port TCP > 1023 et plusieur Client peux connecter avec notre serveur FTP grace l'utilisation des Threads différents.

notre serveur devait au moin capable de gérer les commandes : a avoir des fonctionnalité avec les command :

- USER

- PASS

- RETR

- STOR

- LIST

- QUIT

- PWD

- CWD

- CDUP

1. **Command :**

**USER <Name User> :** Authentification utilisateur , Vérifier si se utilisateur à un droit de connecter avec notre serveur FTP, action nécessaire pour se connecter avec notre serveur FTP

**PASS <Password > :** Mode de passe d’utilisateur ,pour Confirmer l'authentification, cette action est nécessaire avant d’éxecuter une action sur notre serveur .

**QUIT :** Command utiliser quand Client souhaite quitter et finir la connexion avec le servuer FTP.

**Aprés le client et Réussie l’authentification :**

**RETR <nom du fichier> :** Cette Command permet de récupérer un fichier présent sur le serveur FTP qui porte porte le nom du fichier dans le répertoire courant distant

**STOR <nom du fichier> :** Cette Command permet d’envoyer un fichier local qui porte le nom du fichier sur le serveur FTP.

**LIST :** Cette Command permet de lister les fichiers et dossiers présents dans le répertoire courant distant.

**PWD :** Cette Command afficher le nom complet du répertoire courant distant.

**CWD <chemin> :** Cette Command permet de changer le répertoire courant distant .

**CDUP :** cette Command permet d'accéder au répertoire parent

1. **Exécution Programme :**

La conception de Serveur Ftp et fait à partir de logiciel NetBean ,et pour le client j’ai utiliser « Command Ftp » de Win10 .

Le Serveur Ftp écoute et attend une connexion sur le port 1069, une fois une connexiion et établie avec notre serveur FTP , le serveur créer un nouveau Thread qui s’occupe du client qui vient de connecter.

Aprés notre serveur mettre à nouveau l’écoute sur le port 1069 en attente une nouvelle connexion.

Aprés un Thread et Créer il gére toutes les commandes que le client lui envoie, jusqu’à ce que se client décide de quitter et finir la connexion.

Notre Serveur est capable de focntionner avec la command ftp (Win10 ) FIlezilla.

1. **Architecture Utilser :**
2. **Packages Server :**
   1. FtpRequest :

La classes qui traite un ensemble des focntionnalité de serveur ,ainsi gére les connexion de serveur passif et actif ,ainsi tout les command vas etre éxécuter par notre client.

* 1. Server :

La classe qui contient le Main().

Sur cette class une boucle infinie et lancé en attendant que des client se connecter sur notre sereur FTP,pour chaque connexion notre serveur créer une thread .

1. **Package Exception :**

Sur se package on définie un ensemble des exception qui sera déclenchées , pour un erreur de l’authentification , ou demande d’accées a un répertoir non autorisée , ainsi l’éxécution d’une command non correct.

1. **Package Command :**

Dans se package on 3 class :

* 1. Command :

Dans cette class on définie un ensemble des messages qui sera envoyer par notre serveur ou notre client , pour informer notre client de la réussite ou échec d’une focntionnalité.

* 1. Difinition server :

cette class pour définir le numero de port , repertoir ,repertoir courant de notre serveur FTP,ainsi utiiser pour récupérer le repertoir courant et repertoire de notre serveur.

* 1. UserClass :

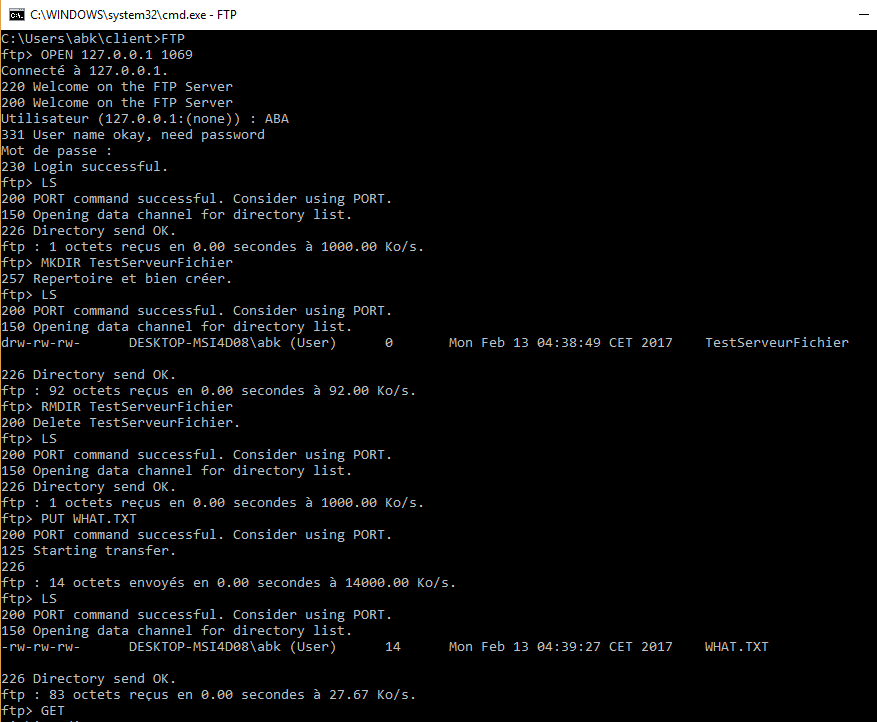
Class pour gérer et créer des utilisatuer pour notre serveur FTP.

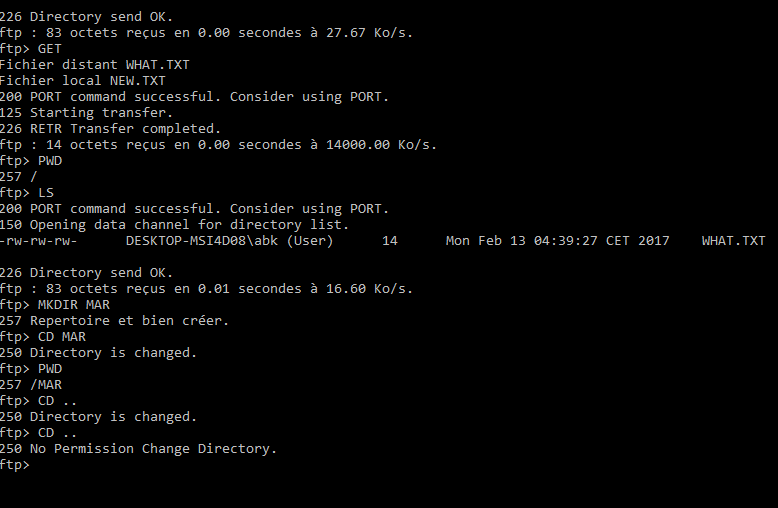
1. **Package Test :**

La classe FTPRequestTest contient des tests unitaires réalisés à l'aide de JUnit et Mockito.

Juint et Mockito nos permet de faire des test pour vérifier le bon fonctionnement de nos méthode qui sera éxécuter par notre serveur.

1. **Connected to Client :**





1. **Uml :**

UML diagramme dans UML.png