

LEFAIST Tristan, disk.php (voir en bas pour les commandes)

GERER LES UTILISATEURS POUR DISK.PHP

=====

- Précréer un répertoire USERS (C:/wamp64n/www/USERS/)
- Chaque utilisateur correspond à un répertoire dans USERS
- Connexion d'un user avec son nom et son mot de passe
 - o Vérifier dans la BD que l'utilisateur existe ...
 - o Créer une session (espace mémoire associé à un user sur un serveur qui disparaît au bout d'un certain temps) avec :
 - le nom de l'utilisateur (ex. DUPONT)
 - Le répertoire racine de l'utilisateur (C:/wamp64n/www/USERS/DUPONT)
 - Le répertoire courant relatif à la racine de l'utilisateur (ex. plans/appartement/ ==> C:/wamp64n/www/USERS/DUPONT/plans/appartement/)

SYNTAXE DES COMMANDES

=====

HTTP (POST)

PARAM1 =USERNAME / PARAM2 = PASSWORD

LOGIN UserName Password CMD=LOGIN & PARAM1 & PARAM2=Chapeau

LOGOUT CMD=LOGOUT

DIR CMD=DIR

CD Plans CMD=CD & PARAM1=Plans

TARGET=PARAM1

REPONSES STANDARDISEES

=====

JSON

```
{
  cmd:"DIR",          // Rappel de la commande envoyée
  status:OK,          // Statut : Erreur ou OK
  content:             // Contenu de la réponse
  [
    { type:DIR ,       name:Divers  }    // un répertoire
    { type:FILE, size:13456 ,   name:plan.doc }    // un fichier
    { type:FILE, size:789   ,   name:info.txt  }
  ]
}

{
  cmd:"CD Plans",
  status:"OK",
  content: null
}
```

Déterminer la liste des commandes possibles avec le disk.manager.

Expliquer pour chaque commande, son rôle et son effet sur le disque.

Donner un exemple de syntaxe Http de la commande, avec ses paramètres.

Donne un exemple de syntaxe JSON de la réponse à cette comande.

COMMANDES

Dossier :

newDir(nom)

—> Nouveau dossier

delDir(nom)	—> Supprimer dossier
Lister(nom)	—> Lister les dossiers et fichiers

Fichier :

newFile(nom)	—> Nouveau fichier
writeFile(nom, content)	—> Ecrire dans le fichier
readFile(nom)	—> Lire le fichier
delFile(nom)	—> supprimer fichier

Session :

Login(user, mdp)	—> se connecter
Logout	—> se déconnecter

CD —> Changement de répertoire

moove —> déplacement fichier

Cp —> copier fichier

-r —> récursivité

Tache automatique

Exemple Syntaxe HTTP :

Content-Type: text/html

Content-Length: 123

Date: Lun, 11 Oct 2023 12:00:00 GMT

<!DOCTYPE html>

<html>

<head>

<title>Exemple de page</title>

</head>

<body>

```
<h1>Ceci est une page d'exemple</h1>
<p>Contenu de la page.</p>
</body>
</html>
```

Méthode GET :

GET /exemple/page.html HTTP/1.1

Host: www.example.com

Accept: text/html

User-Agent: MonAgentUtilisateur

Syntaxe JSON de la réponse :

```
{
  "status": "success",
  "message": "La demande a été traitée avec succès" ou "OK",
  "data": {
    "user_id": 12345,
    "username": "utilisateur123",
    "email": "utilisateur123@example.com"
  }
}
```