

	Faculté des Sciences de Sfax Département d'informatique et des Communications	Année Universitaire 2025 – 2026 Auditoire LSI 3
	Matière : Développement d'applications réparties Responsable du cours : Corinne Amel ZAYANI Enseignantes des TP : Fatma ELLOUZE, Zeineb DAOUD	

TP 3 : Serveur Multi-threads - sockets en mode connecté TCP

Activité 3-1 : Acceptation de plusieurs clients

Objectifs

Cette activité vise à :

- Comprendre et appliquer le principe d'un serveur multithread basé sur les sockets TCP en Java.

Environnement

- JDK ≥ 1.8
- Eclipse IDE for Java developers Ou IntelliJ IDAE > 1.8 OU VS code
- Git

Type de l'activité

- Individuel

Ressources

- Consultez le cours "Développement d'applications réparties" : **séances 2, 3 et 4** (CC / objet / serveur multi-thread), 03/10 - 10/10 - 17/10//2025.
- Applications Client/Serveur développées dans les **TP1** et **TP2**.

Livrables

- Le lien du repository dans l'espace de dépôt de la plateforme ENT
- Le fichier READ ME

Enoncé

Les deux applications, client et serveur, développées jusqu'à présent (dans les TP1 et TP2), ne permettent de gérer qu'un seul client à chaque exécution du serveur. Nous souhaitons désormais les modifier afin de concevoir un véritable serveur capable de traiter plusieurs clients simultanément.

Travail demandé

Dans cette activité, il vous est demandé d'améliorer le squelette de communication Client/Serveur afin de permettre la connexion simultanée de plusieurs clients à un même serveur en utilisant les sockets TCP.

Cette activité comporte trois étapes :

Étape 1 : Préparation de l'espace de travail

1. Dans votre espace de travail, créez un répertoire sous le nom « **votreNomPrenom_LSI3_DevAppReparties_TP3** ».
2. Créez un document Word nommé « **VotreNomPrenom_Activite_3_1.docx** » pour l'insertion des captures d'écran de chaque étape réalisée.

Étape 2 : Création d'un serveur multi-thread

Dans cette étape, vous êtes demandé de créer un nouveau projet Java nommé « TP3 », dans lequel vous allez étendre le squelette de communication Client/Serveur développé dans l'**activité 2-1**. Pour ce faire, vous devez suivre ces phases :

1. Implémentation

Vous devez réaliser les traitements suivants :

- a) Développer un serveur multi-thread capable de gérer plusieurs clients simultanément.
- b) Afficher, côté serveur, les informations de chaque client connecté, notamment :
 - son adresse IP,
 - et son numéro d'ordre de connexion (client n°1, client n°2, etc.).

NB : Pour récupérer l'adresse IP d'un client connecté, vous pouvez utiliser la méthode `getRemoteSocketAddress()` de la classe `Socket`.

- c) Envoyer à chaque client son numéro d'ordre de communication.

2. Tests locaux

Vérifiez la connexion entre le serveur et plusieurs clients sur votre machine en utilisant :

- a) Des clients Telnet¹ : ouvrez une invite de commande et tapez → *telnet localhost 1234*
- b) Des clients Java que vous avez développés.

Remarque :

Telnet n'est pas toujours activé par défaut. Vous pouvez l'activer via :

Panneau de configuration → Programmes → Activer ou désactiver des fonctionnalités
Windows → Client Telnet

¹ **Telnet** (Telecommunication Network) est à la fois un protocole de communication et un outil en ligne de commande permettant d'établir une connexion à distance avec un autre ordinateur ou un serveur via un réseau TCP/IP. Il offre un accès distant en mode texte (ligne de commande) et permet d'interagir directement avec le serveur.

Parmi ses principales fonctionnalités, Telnet permet d'établir une connexion client/serveur, de transmettre les commandes saisies par l'utilisateur au serveur distant et d'afficher les réponses de ce dernier sur le terminal local.

3. Tests en réseau local

Testez ensuite la connexion entre plusieurs machines du même réseau, en configurant l'une comme serveur et les autres comme clients.

Pour cela, adaptez le code des classes Client en utilisant les classes **InetAddress** et **InetSocketAddress** afin de spécifier l'adresse IP du serveur distant.

Étape 3 : Dépôt des livrables

Dans cette dernière étape, vous êtes invités à :

- Déposez votre projet sur GitHub ainsi que le lien du repository dans l'espace de dépôt de la plateforme ENT, et le mentionnez dans le document Word.
- Déposez votre document Word dans l'espace de dépôt sur l'ENT :
 - Ajoutez également le lien vers votre dépôt GitHub.
 - Le document doit contenir des imprimés écrans accompagnés d'une synthèse.