

## SRCS : Systèmes Répartis Client/Serveur

### API socket Java

Julien Sopena

Janvier 2012

L'objectif de ce TP est de mettre en oeuvre l'API socket du langage Java au travers de la programmation d'un serveur Web.

#### Exercice 1 : Serveur Web avec commandes *GET* et *PUT*

##### Question 1

Écrire un programme Java faisant office de serveur Web et traitant seulement les commandes *GET*. Pour cela, écrire les deux classes suivantes :

1. une classe *Serveur* avec une méthode *main* :
  - écoutant les demandes de connexion sur un port *TCP* à 1023 ;
  - déléguant à l'aide d'un thread le traitement d'une requête entrante à un objet de la classe *HttpRequest*.
2. une classe *HttpRequest* comportant :
  - une méthode *processRequest* des traitements généraux concernant une requête entrante et déléguant le traitement de la commande *GET* à la méthode *processGet* ;
  - une méthode *processGet* se chargeant de traiter la commande *GET*.

Afin de pouvoir lire simplement ligne à ligne les données d'un *InputStream*, vous pourrez utiliser la classe *InputStreamLiner*. Le programme suivant illustre son fonctionnement en lisant ligne à ligne les données d'une socket.

```
class Exemple () {  
    void exemple() {  
        InputStream is = aSocket.getInputStream();  
        InputStreamLiner isl = new InputStreamLiner(is);  
        String s;  
        while ( (s=isl.readLine(InputStreamLiner.UNIX)) != null ) {  
            System.out.println(s);  
        }  
        isl.close();  
    }  
}
```

**Exercice 2 : Serveur Web avec commande GET****Question 1**

Modifier le serveur Web précédent pour prendre en compte les commandes PUT. Ajouter entre autre, une méthode *processPut* à la classe *HttpRequest*.

**Question 2**

Écrire un programme client qui se connecte sur le serveur Web et lui envoie une commande *PUT*.

**Exercice 3 : Serveur Web avec commandes GET, PUT et synchronisation****Question 1**

Étendre le serveur Web précédent pour faire en sorte que les lectures et les écritures simultanées sur un même fichier se fassent de manière cohérente.