

# TD Internet et Réseaux partie II

## Web, protocole HTTP sur des sockets

Ce TD a pour but de vous faire développer votre propre serveur Web que vous pourrez tester avec un simple telnet localhost 8080 puis via un navigateur Web standard. Vous pourrez vous aider pour cela des informations du cours et/ou celles de :

<http://www.commentcamarche.net/internet/http.php3>

### 1 Rappel sur les sockets

Vous trouverez les principales fonctions de l'API/C# Socket sur <http://profdinfo.com/web/420-KHH-LG/sockets.html>.

Vous trouverez une série d'exemples détaillés sur <http://www.java2s.com/Code/CSharpAPI/System.Net.Sockets/SocketConnectIPEndPointip.htm>.

Les tests se feront grâce la commande telnet que vous installerez si elle n'est pas encore présente sur votre machine.

Vous trouverez toutes les explications pour cela sur le site du cours sur [www.tigli.fr](http://www.tigli.fr).

### 2 Description de « l'architecture » d'un serveur Web

Pour vous aider voici quelques informations et questions :

1. Un serveur Web est un serveur socket mode connecté (TCP/IP). Vous pourrez donc utiliser au choix Java ou C/C++ pour vos développements.
2. Le port standard pour un serveur WEB est le port 80. Pourquoi devons nous utiliser ici le port 8080 ?
3. Après avoir créé la connexion TCP/IP un serveur dialogue avec un client en utilisant le protocole HTTP (Cf. cours et/ou RFC 1945).

### 3 Tests avant de commencer

Commencer par tester à l'aide de la commande telnet la mise en place d'un dialogue avec un serveur existant.

Pour régler votre telnet sous windows, utilisez les informations de la pages <http://windows.microsoft.com/fr-fr/windows/telnet-commands#1TC=windows-7>. En particulier

Vous pourrez par exemple tester la commande suivante (au cas où, installez telnet sur votre machine) :

```
telnet
set LOCALECHO
open www.google.fr 80
GET /
```

les informations qui vous sont retournées sont au format HTML, celui des pages Web !

## TD Internet et Réseaux partie II

### Web, protocole HTTP sur des sockets

---

Vous pouvez de même installer Google Chrome et utilisé "les outils de développement" du browser Web. et voir combien de requêtes HTTP sont déclenchées pour récupérer l'intégralité de la page.

#### 4 Mise en œuvre d'un serveur HTTP basique

Commencer par créer un serveur en C# en vous inspirant du code source qui vous est fourni sur les pages web du cours (Serveur Socket Base HTTP - au format TXT à convertir en CS.rtf à convertir en Serveur Socket Base HTTP.cs).

Lisez les consignes dans le fichier pour le modifier, en vous référant au cours sur le protocole HTTP.

Votre serveur doit prendre en compte une partie simple du protocole HTTP du type :

- *GET <chemin relatif d'un fichier HTML par rapport à la racine du site Web>*
- *Exemple GET /index.html http/1.0, correspond pour le serveur à l'envoi du fichier /www/pub/index.html*

Voici les données que vous pouvez renvoyer.:

```
<TITLE> L'exemple HTML le plus simple</TITLE>
<H1> Ceci est un sous-titre de niveau 1</H1>
Bienvenue dans le monde HTML. Ceci est un paragraphe. <P>
Et ceci en est un second. <P>
<A HREF="index.html">cliquez ici</A> pour réafficher
```

Les plus avancés pourront les déposer dans un fichier qui sera ouvert et lu pour le renvoi des données. C'est ainsi que marche le Web. Vous pourrez déclarer une variable HTTP\_ROOT qui permette de spécifier le chemin d'accès sur votre machine à l'espace disque qui sera servi par votre serveur Web (répertoire racine des documents accessibles par le serveur).