



## TD1 : Prise en main (les servlets)

### Cas d'utilisation 1 :

Créer une servlet qui affiche le message suivant : « **my name is X** ». Le contenu « X » doit être fourni dans l'URL.

### Cas d'utilisation 2 :

Créer une servlet qui affiche le contenu suivant :

```
requestURL: http://localhost:8080/ServletTutorial/other/exampleInfo
requestURI: /ServletTutorial/other/exampleInfo
contextPath: /ServletTutorial
servletPath: /other/exampleInfo
queryString: text1=Hello&text2=World
getParameter text1: Hello
getParameter text2: World

Server info:
serverName: localhost
serverPort: 8080

Client info:
remoteAddr: 0:0:0:0:0:0:1
remoteHost: 0:0:0:0:0:0:1
remoteHost: 63302
remoteUser: null

headers:
host: localhost:8080
connection: keep-alive
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
user-agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
accept-encoding: gzip, deflate, sdch
accept-language: vi-VN;q=0.8,fr-FR;q=0.6,fr;q=0.4,en-US;q=0.2,en;q=0.2
alexatoolbar-abx_ns_ph: AlexaToolbar/alxg-3.3

Servlet Context info:
realPath: F:\ECLIPSE_TUTORIAL\SERVLET\metadata\plugins\org.eclipse.wst.server.core\tmp1\wtpwebapps\ServletTutorial\
```

### Cas d'utilisation 3 :

- Créer un document HTML qui affiche le formulaire suivant :

User Name:   
Password:   
-----  
First Name:   
Last Name:   
Gender:  Male  Female  
-----  
Address:  
  
  
  
-----

- Créer une servlet qui récupère les données du formulaire et les affiche après avoir validé l'entrée User Name qui doit être un email valide. NB : utiliser les **expressions régulières**.

### Cas d'utilisation 4 :

Créer une application qui simule un panier électronique :

- En utilisant les cookies
- En utilisant les sessions

L'inscription (**authentification**) à l'application devra passer par un **filtre**. On suppose que les login et mots de passe sont enregistrés dans un fichier « **messages.properties** ».



### **Cas d'utilisation 5:**

Reprenez le cas d'utilisation n°3 ; ce qui est demandé est de retarder l'affichage des données 5 secondes. En effet, la servlet devra supporter le mode **asynchrone** pour traiter la requête.

```
@WebServlet(urlPatterns= "/CatalogServlet", asyncSupported = true)
```

Penser à utiliser l'objet **AsyncContext**.