

## *Rapport de Tp1*

Prise en main de l'environnement JEE dans Apache  
Tomcat

Realisé par:

Salma fassi

G.Info 2

Encadré par:

Pr. Youness IDRISSE

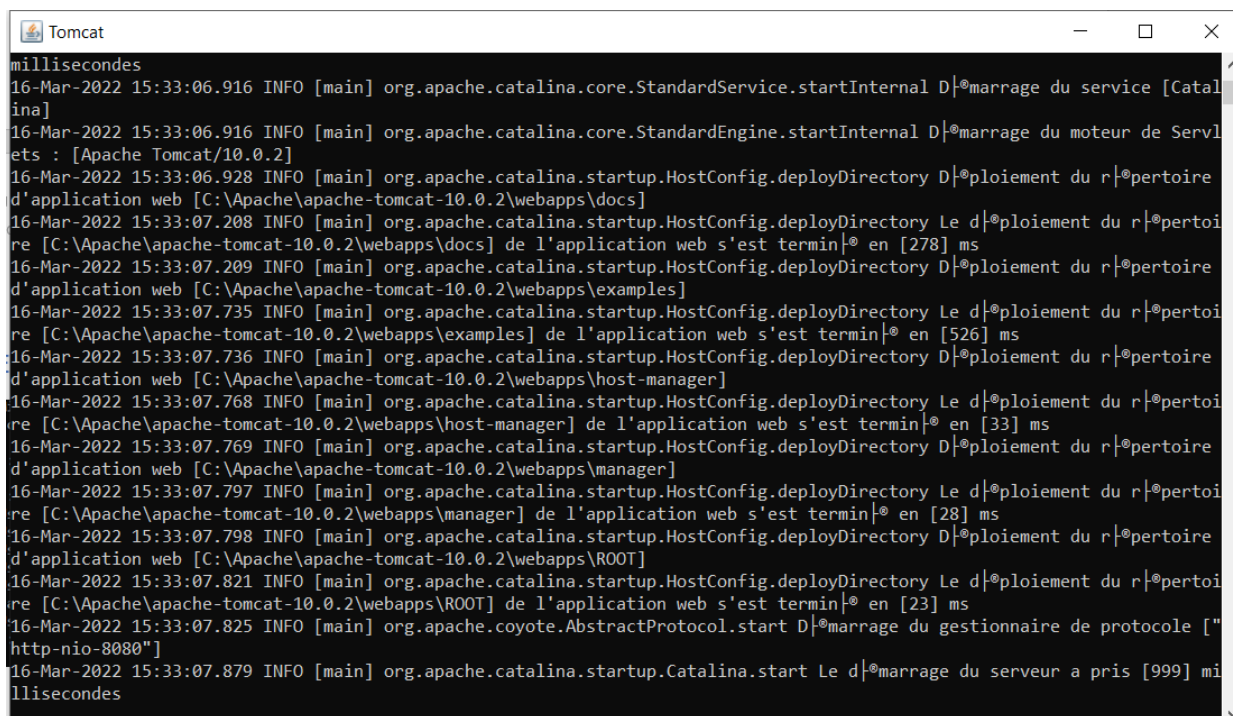
KHAMLIHI

## Exercice 1 :

### 1. Lancez Tomcat startup.bat dans le répertoire TOMCAT\_HOME/bin :

```
C:\Users\lenovo>cd C:\Apache\apache-tomcat-10.0.10\bin

C:\Apache\apache-tomcat-10.0.10\bin>startup.bat
Using CATALINA_BASE:   "C:\Apache\apache-tomcat-10.0.10"
Using CATALINA_HOME:   "C:\Apache\apache-tomcat-10.0.10"
Using CATALINA_TMPDIR: "C:\Apache\apache-tomcat-10.0.10\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk-17.0.1"
Using CLASSPATH:       "C:\Apache\apache-tomcat-10.0.10\bin\bootstrap.jar;C:\Apache\apache-tomcat-10.0.10\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
```



Tomcat

```
millisecondes
16-Mar-2022 15:33:06.916 INFO [main] org.apache.catalina.core.StandardService.startInternal D:\marrage du service [Catalina]
16-Mar-2022 15:33:06.916 INFO [main] org.apache.catalina.core.StandardEngine.startInternal D:\marrage du moteur de Servlets : [Apache Tomcat/10.0.2]
16-Mar-2022 15:33:06.928 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory D:\p\ploiment du r\p\ertoire d'application web [C:\Apache\apache-tomcat-10.0.2\webapps\docs]
16-Mar-2022 15:33:07.208 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Le d\p\ploiment du r\p\ertoire [C:\Apache\apache-tomcat-10.0.2\webapps\docs] de l'application web s'est termin\ en [278] ms
16-Mar-2022 15:33:07.209 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory D\p\ploiment du r\p\ertoire d'application web [C:\Apache\apache-tomcat-10.0.2\webapps\examples]
16-Mar-2022 15:33:07.735 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Le d\p\ploiment du r\p\ertoire [C:\Apache\apache-tomcat-10.0.2\webapps\examples] de l'application web s'est termin\ en [526] ms
16-Mar-2022 15:33:07.736 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory D\p\ploiment du r\p\ertoire d'application web [C:\Apache\apache-tomcat-10.0.2\webapps\host-manager]
16-Mar-2022 15:33:07.768 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Le d\p\ploiment du r\p\ertoire [C:\Apache\apache-tomcat-10.0.2\webapps\host-manager] de l'application web s'est termin\ en [33] ms
16-Mar-2022 15:33:07.769 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory D\p\ploiment du r\p\ertoire d'application web [C:\Apache\apache-tomcat-10.0.2\webapps\manager]
16-Mar-2022 15:33:07.797 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Le d\p\ploiment du r\p\ertoire [C:\Apache\apache-tomcat-10.0.2\webapps\manager] de l'application web s'est termin\ en [28] ms
16-Mar-2022 15:33:07.798 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory D\p\ploiment du r\p\ertoire d'application web [C:\Apache\apache-tomcat-10.0.2\webapps\ROOT]
16-Mar-2022 15:33:07.821 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Le d\p\ploiment du r\p\ertoire [C:\Apache\apache-tomcat-10.0.2\webapps\ROOT] de l'application web s'est termin\ en [23] ms
16-Mar-2022 15:33:07.825 INFO [main] org.apache.coyote.AbstractProtocol.start D\marrage du gestionnaire de protocole ["http-nio-8080"]
16-Mar-2022 15:33:07.879 INFO [main] org.apache.catalina.startup.Catalina.start Le d\marrage du serveur a pris [999] millisecondes
```

## 2. le fichier server.xml :

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
<!-- Note: A "Server" is not itself a "Container", so you may not
      define subcomponents such as "Valves" at this level.
      Documentation at /docs/config/server.html
-->
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!-- APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
```


La balise principale	<code>&lt;Server port="8005" shutdown="SHUTDOWN"&gt;</code>
Les balises filles et le nombre de leurs apparitions	Listner, GlobalNamingRsources, Resource, Service, Engine, Engine, Realm, Host.
La balise connector et ces attributs	<code>&lt;connector executor= "tomcat ThreadPool" port="8080" protocol="Http/1.1" connectionTimeout="20000" redirectPort="8443" /&gt;</code>

## 3. le fichier « tomcat-users.xml » :


[Aaps](#) [Tout ce que vous d...](#) [YouTube](#) [Développons en Ja...](#) [Class diagrams - Ja...](#) [Programmation ori...](#) [10 best books to le...](#) [Base du langage C#...](#)

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/10.0.10

 SOFTWARE FOUNDATION  
http://www.apache.org

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:  
[Security Considerations How-To](#)  
[Manager Application How-To](#)  
[Clustering/Session Replication How-To](#)

Server Status  
Manager App  
Host Manager

Developer Quick Start

[Tomcat Setup](#)  
[First Web Application](#)

[Realms & AAA](#)  
[JDBC DataSources](#)

[Examples](#)

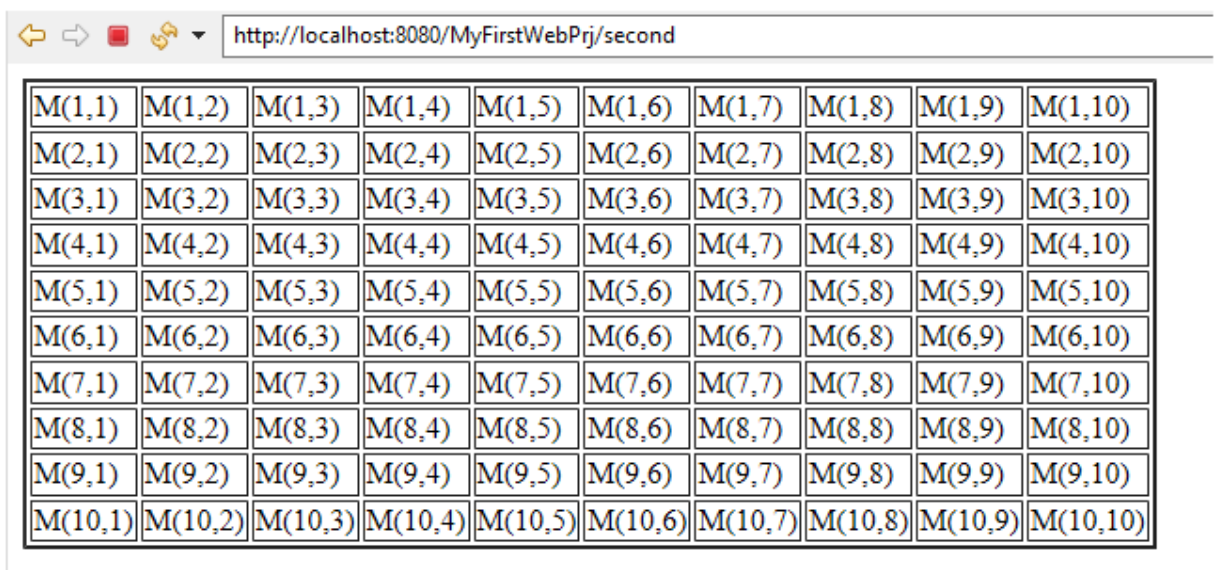
[Servlet Specifications](#)  
[Tomcat Versions](#)

## Exercice 2 :

### 1. un tableau de 10 cases par 10 :

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3
4 import javax.sql.rowset.serial.SerialException;
5
6 import jakarta.servlet.http.HttpServlet;
7 import jakarta.servlet.http.HttpServletRequest;
8 import jakarta.servlet.http.HttpServletResponse;
9
10 public class servlet1 extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12
13     /**
14      * Default constructor.
15      */
16
17     /**
18      * @throws IOException
19      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
20      */
21     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
22         // TODO Auto-generated method stub
23         //response.getWriter().append("Served at: ").append(request.getContextPath());
24
25         response.setContentType("text/html");
26
27         PrintWriter out = response.getWriter();
28         out.println("<table border='2'>");
29         for(var i=1; i<11; i++) {
30             out.println("<tr>");
31             for(var j=1; j<11; j++) {
32                 out.println("<td>" + "M(" + i + "," + j + ")" + "</td>");
33             }
34             out.println("</tr>");
35         }
36         out.println("</table>");
37     }
38
39     /**
40      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
41      */
42     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
43         // TODO Auto-generated method stub
44         doGet(request, response);
45     }
46 }
```

#### ➤ Affichage :



The screenshot shows a web browser window with the address bar displaying "http://localhost:8080/MyFirstWebPrj/second". The main content area displays a 10x10 table with a border. Each cell in the table contains a string representing a coordinate, "M(i,j)", where i is the row index and j is the column index, both ranging from 1 to 10. The table is rendered with a light blue background and black borders.

M(1,1)	M(1,2)	M(1,3)	M(1,4)	M(1,5)	M(1,6)	M(1,7)	M(1,8)	M(1,9)	M(1,10)
M(2,1)	M(2,2)	M(2,3)	M(2,4)	M(2,5)	M(2,6)	M(2,7)	M(2,8)	M(2,9)	M(2,10)
M(3,1)	M(3,2)	M(3,3)	M(3,4)	M(3,5)	M(3,6)	M(3,7)	M(3,8)	M(3,9)	M(3,10)
M(4,1)	M(4,2)	M(4,3)	M(4,4)	M(4,5)	M(4,6)	M(4,7)	M(4,8)	M(4,9)	M(4,10)
M(5,1)	M(5,2)	M(5,3)	M(5,4)	M(5,5)	M(5,6)	M(5,7)	M(5,8)	M(5,9)	M(5,10)
M(6,1)	M(6,2)	M(6,3)	M(6,4)	M(6,5)	M(6,6)	M(6,7)	M(6,8)	M(6,9)	M(6,10)
M(7,1)	M(7,2)	M(7,3)	M(7,4)	M(7,5)	M(7,6)	M(7,7)	M(7,8)	M(7,9)	M(7,10)
M(8,1)	M(8,2)	M(8,3)	M(8,4)	M(8,5)	M(8,6)	M(8,7)	M(8,8)	M(8,9)	M(8,10)
M(9,1)	M(9,2)	M(9,3)	M(9,4)	M(9,5)	M(9,6)	M(9,7)	M(9,8)	M(9,9)	M(9,10)
M(10,1)	M(10,2)	M(10,3)	M(10,4)	M(10,5)	M(10,6)	M(10,7)	M(10,8)	M(10,9)	M(10,10)

## 2. envoyer la réponse vers un fichier Excel :

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.sql.rowset.serial.SerialException;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class servlet1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */

    /**
     * @throws IOException
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // TODO Auto-generated method stub
        //response.getWriter().append("Served at: ").append(request.getContextPath());

        response.setContentType("application/vnd.ms-excel");
        response.setHeader("Content-Disposition", "attachment;filename=MaServlet.xls");
        PrintWriter out = response.getWriter();
        out.println("<table border='2'>");
        for(var i=1; i<11; i++) {
            out.println("<tr>");
            for(var j=1; j<11; j++) {

                out.println("<td>"+ "M("+i+", "+j+")"+"</td>");
            }
            out.println("</tr>");
        }
        out.println("</table>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```

➤ *Affichage :*

[illegible]

## Exercice 3 :

### 1. Récupération des informations sur l'url :

```
import java.io.IOException;
import java.io.PrintWriter;

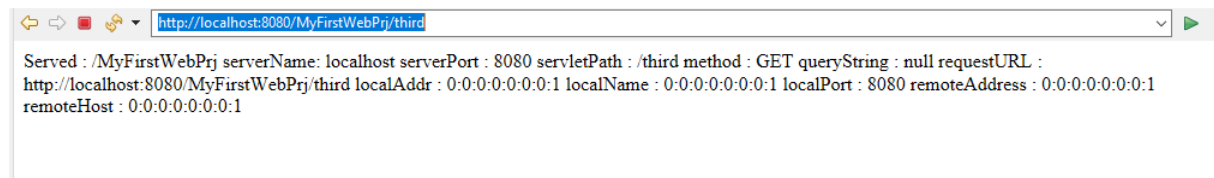
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
public class servlet2 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        //response.getWriter().append("Served at: ").append(request.getContextPath());

        PrintWriter out= response.getWriter();

        out.println("Served : " + request.getContextPath());
        out.println("serverName: " + request.getServerName());
        out.println(" serverPort : " + request.getServerPort());
        out.println(" servletPath : " + request.getServletPath());
        out.println(" method : " + request.getMethod());
        out.println(" queryString : " + request.getQueryString());
        out.println(" requestURL : " + request.getRequestURL());
        out.println(" localAddr : " + request.getLocalAddr());
        out.println(" localName : " + request.getLocalName());
        out.println(" localPort : " + request.getLocalPort());
        out.println(" remoteAddress : " + request.getRemoteAddr());
        out.println(" remoteHost : " + request.getRemoteHost());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```

#### ➤ Affichage :



### 2.

#### a) Le formulaire en JSP :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method='post' action='formulaire' >
<input type='text' name='nom' placeholder='nom' />
<input type='text' name='prenom' placeholder='prenom' />
<input type='submit' value='connexion'>
<input type='reset' value='reset'>
</form>
</body>
</html>
```

- *Le servlet pour traiter le formulaire :*

```
import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

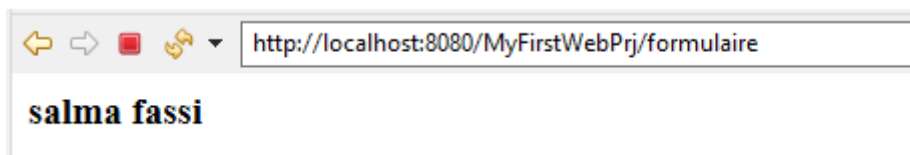
public class servletform extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException {
    }
    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {
        String nom=req.getParameter("nom");
        String prenom=req.getParameter("prenom");
        res.setContentType("text/html");
        PrintWriter out=res.getWriter();
        if(nom.length()!=0 && prenom.length()!=0) {
            out.print("<h3>"+nom+"| "+prenom+"</h3>");
        }
        else {
            out.print("veuillez remplir");
        }
    }
}
```

- *L'exécution :*

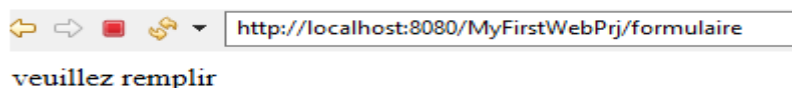
*Le formulaire :*



*Cas de succès :*



*Cas d'échec :*





b) *Le longueur de texte récupéré :*

1. *En utilisant `getContentLength()` :*

*Pour récupérer la longueur de texte, on utilise La fonction `getContentLength()` :*

```
out.print("la longueur de texte est "+req.getContentLength());
```

*Résultat :*

**salma fassi**

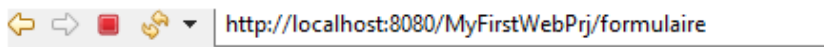
la longueur de texte est 22

2. *En utilisant `length()` :*

*Si on applique la propriété `length()` sur le champ nom et prénom*

```
out.print("la longueur de texte est "+(nom+prenom).length());|
```

*Résultat :*



**salma fassi**

la longueur de texte est 10

➤ *Conclusion : la différence entre deux méthodes est gigantesque*

c) *Après qu'on copie le code source, on obtient plus de détails concernant l'entête et ses contenus.*



### 3. Affichage des informations sur client :

La page servlet « echo.java » :

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class echo extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException {
        PrintWriter out=res.getWriter();
        out.println("<h1>Servlet Echo</h1><br>");
        out.println("Request method: "+req.getMethod()+"</br>");
        out.println("Request url: "+req.getRequestURL()+"</br>");
        out.println("Request url: "+req.getProtocol()+"</br>");
        Enumeration<String> headerNames=req.getHeaderNames();
        out.println("<h2><center>les entetes provenant du client</center></h2>");
        out.println("<table border=2>");
        out.println("<tr><th ><Header Name/th><th>Header Value</th></tr>");
        while(headerNames.hasMoreElements()) {
            String name=headerNames.nextElement();
            String value=req.getHeader(name);
            out.println("<tr><th>"+name+"</th><th>"+value+"</th></tr>");
        }
        out.println("</table>");
    }
    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {
        doGet(req,res);
    }
}
```

➤ L'exécution :

## Servlet Echo

Request method: GET  
Request url: http://localhost:8080/MyFirstWebPrj/echo  
Request url: HTTP/1.1

### les entetes provenant du client

	Header Value
accept	image/gif, image/jpeg, image/pjpeg, application/x-ms-application, application/xhtml+xml, application/x-ms-xbap, */*
accept-language	en-US,en;q=0.91,fr-FR;q=0.82,fr-BE;q=0.73,fr;q=0.64,es-ES;q=0.55,es;q=0.45,ar-MA;q=0.36,ar-SA;q=0.27,ar;q=0.18,qaa-Latn;q=0.091
cache-control	no-cache
ua-cpu	AMD64
accept-encoding	gzip, deflate
user-agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident/7.0; rv:11.0) like Gecko
host	localhost:8080
connection	Keep-Alive
cookie	JSESSIONID=9C44939842A0A7D8880A8E8575A75F12

#### Exercice 4 :

##### 1. L'affichage de factorielle de nombre entre 0 et 9 :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
        return i*fact(i-1);
    }
}
%>
<%
for(int x=0;x<10;x++)
{
%>
<h3><%= fact(x) %></h3>
<% } %>
</body>
</html>
```

➤ L'exécution :

1

1

2

6

24

120

720

5040

40320

362880

## 2. Le factorielle de nombre entre 0 et n :

Le fichier JSP :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
        return i*fact(i-1);
    }
}
%>
<form method="get">
<input type="number" name="nbr" value="" placeholder="entrez un integer">
<input type="submit" value="factoriel">
</form>
<%
if(request.getParameter("nbr")!=null){
    int number=Integer.parseInt(request.getParameter("nbr"));
    %>
    <%
    for(int i=0;i<=number;i++){
    %>
    <h3><%= i %>! = <%= fact(i) %></h3>
    <%>
    }
    %>
</body>
</html>
```

➤ L'exécution :

3	factoriel
---	-----------

0! = 1

1! = 1

2! = 2

3! = 6

### 3. Le factoriel d'un nombre lu au clavier :

Le fichier JSP

```
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
        return i*fact(i-1);
    }
}
%>
<%
if(request.getParameter("nbr")!=null){
    int number=Integer.parseInt(request.getParameter("nbr"));

%>
<form method="get">
<input type="number" name="nbr" value=<%= number%> placeholder="entrez un integer">
<input type="submit" value="factoriel">
</form>
<%}
if(request.getParameter("nbr")!=null){
    int number=Integer.parseInt(request.getParameter("nbr"));

%>

<h3><%= number %>! = <%= fact(number) %></h3>
<%}
%>
```

➤ L'exécution :

3	factoriel
---	-----------

**3! = 6**