

Exercice 1 et 2

L'exercice 1 nous apprend la différence entre une page .html et une page .jsp. Une page html est statique (sans JS) et rendered par le browser. Au contraire, une page jsp est, en premier lieu, traitée par un serveur TomCat qui va exécuter le code présent dans la page, permettant d'avoir un contenu dynamique. Voici comment se présente le code Java dans une page jsp.

Voici le fichier index.html :

```
<!--  
Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to  
change this license  
Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Html.html to edit  
this template  
-->  
<html>  
  <head>  
    <title>Hello Word!</title>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  </head>  
  <body>  
    <div>Hello Word!</div>  
  </body>  
</html>
```

Voici le fichier maJsp.jsp :

```
<%--  
  
  Document   : maJsp  
  
  Created on : 23 mars 2023, 14:43:07  
  
  Author    : LeonettiM  
  
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello Word from JSP!</h1>
  </body>
</html>
```

Résultat :



Hello World from JSP!

Exercice 3

Index.html :

```
<form method="post" action="checkLogin.jsp"> <!-- la page checkLogin.jsp
sera appelée lors du submit -->
  <label for="user">Votre nom d'utilisateur :</label>
  <input type="text" name="username" id="user" size="50" maxlength="50"
/><br>
  <label for="pass">Votre mot de passe :</label>
  <input type="password" name="password" id="pass" size="50"
maxlength="50" /><br>
  <input type="submit" value="Soumettre">
</form>
```

Ci-dessus, on crée un form qui va nous permettre d'appeler le fichier checklogin.jsp, qui représente notre page JSP via un bouton.

```
<h1>Contrôle du login par JSP</h1>

<%

    String username = request.getParameter("username");

    String password = request.getParameter("password");

%>

<p>Votre nom d'utilisateur est : <%=username%> <br>

    Votre mot de passe est : <%=password%> <br>

</p>
```

Lorsque l'on appuie sur le bouton on récupère un username et un password pour l'afficher grâce aux scriptlet tags.

Résultat :

Page de login Votre nom d'utilisateur : <input type="text"/> Votre mot de passe : <input type="password"/> <input type="button" value="Soumettre"/>	Contrôle du login par JSP Votre nom d'utilisateur est : admin Votre mot de passe est : Pa\$\$w0rd
---	--

Exercice 4

L'exercice 3 nous démontre la consommation d'un Web Service de météo par une JSP.

Index.html :

```
<form method="post" action="checkMeteo.jsp"> <!-- la page checkLogin.jsp sera appelée lors
du submit -->

    <label for="user">Indiquer un nom d'une ville suisse :</label>

    <input type="text" name="username" id="user" size="50" maxlength="50" /><br>

    <input type="submit" value="Soumettre">

</form>
```

Dans ce code, on appelle grâce à un bouton une page JSP qui va nous permettre d'afficher la météo de la région inscrite dans le textfield .

Code page JSP (index.jsp) :

```
<body>
```

```

<%
    String username = request.getParameter("username");
%>

<h1>Voici la météo de <%=username%> !</h1>

<a href="https://www.prevision-meteo.ch/meteo/localite/<%=username%>">

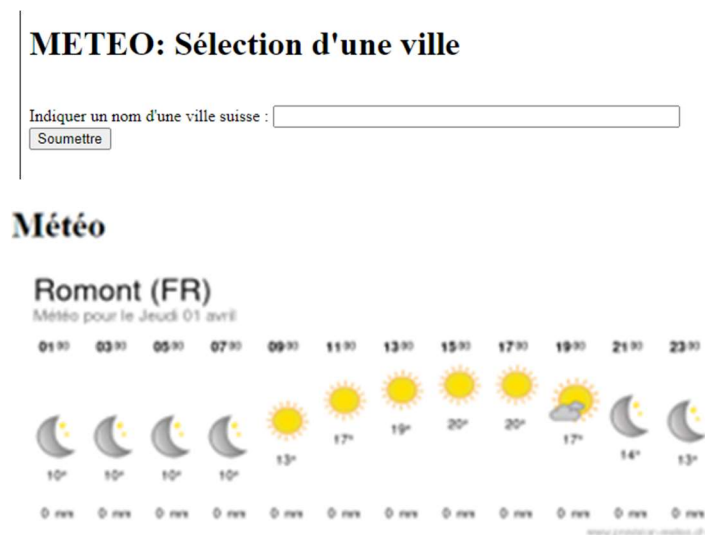
    </a>

</body>

```

On récupère ce que l'on a mis dans notre textfield, et on va le mettre dans la requête qui va nous retourner la météo grâce à une API.

Résultat :



Exercice 5

Ici, j'ai dû afficher des villes qui venait d'une base de données. Il n'y a pas eu de grande difficulté, il a fallu importer la base de données et la mettre en ligne.

Code JSP :

```

<h1>Ma requête SQL depuis JSP!</h1>

<%
    WrkDB wrkDB = new WrkDB("3306", "leonettim_ex5");

    ArrayList<String> lstPays = wrkDB.getPays();

    if (lstPays != null) {
%>

<h2>Liste des pays </h2>

```

```

<%
    out.println("Solution avec instruction java out.println()");
    for (String pays : lstPays) {
        out.print("<div>");
        out.print(pays);
        out.print("</div>");
    }
%>

<div>Solution avec intégration de variables java dans HTML</div>

<%
    for (String pays : lstPays) {
%>
<div><%=pays%></div>
<%

    }

}

%>

```

Il a fallu un foreach pour récupérer les informations, on rajoute ensuite des balises html pour qu'elle vienne à la ligne.

Quand on décide de mettre le site en prod, il faut mettre la base de données en ligne sur cpanel, et changé l'adresse sur laquelle on veut la contacter.

Voici ce qu'il faut changer dans WrkDB :

```

final String url = "jdbc:mysql://5.182.248.183:" + port + "/" + dbName + "?serverTimezone=CET";

final String user = "leonettim_matteo";

final String pw = "VQCgz+PEE_Y[";

boolean result = false;

```

Il faut changer le localhost par mon adresse ip et on change le password et l'utilisateur pour pouvoir accéder à la base de données.

Résultat :

Ma requête SQL depuis JSP!

Liste des pays

Solution avec instruction java out.println()

Suisse

Autriche

France

Slovénie

Allemagne

Canada

USA

Norvège

Italie

Bulgarie

Suède

Solution avec intégration de variables java dans HTML

Suisse

Autriche

France

Slovénie

Allemagne

Canada

USA

Norvège

Italie

Bulgarie

Suède

Exercice 6

On crée deux classes beans qui se nomment « Error » et « Info ». Une sert à afficher les infos de connexion (Info) et l'autre à afficher des erreurs (Error).

Code de Error.java :

```
package com.example.javatt06;

import java.io.Serializable;

private String error;

private String message;

public Error(){

}

public String getError(){
```

```
return error;
}
public String getMessage(){
return message;
}
public void setError(String error){
this.error = error;
}
public void setMessage(String message){
this.message = message;
}
}
```

Info.java :

```
public class Info implements Serializable{
private String username;
private String password;
public Info(){
}
public String getUsername(){
return username;
}
public String getPassword(){
return password;
}
public void setUsername(String username){
this.username = username;
}
public void setPassword(String password){
this.password = password;
}
}
```

Quand on active sur le bouton, on récupère le nom et le mdp. On vérifie ensuite qu'ils correspondent à admin et emf123. On prend ensuite le nom et le mdp et on le forward vers la page JSP « maJSPLogge.jsp ».

```
<html>
<head>
<title>Treatment</title>
</head>
<body>
<%
String username = request.getParameter("username");
String password = request.getParameter("password");
if (username.equals("admin") && password.equals("emf123")) {
Info.setUsername(username);
Info.setPassword(password);
%>
<jsp:forward page="maJspLogge.jsp"></jsp:forward>
}
<%
} else {
Error.setError(request.getRemoteHost());
Error.setMessage("Wrong username or password");
%>
<jsp:forward page="erreur.jsp"/>
<%
}
%>
</body>
</html>
```

Voici le code de la page erreur.jsp :

```
<html>
<head>
<title>Error</title>
```



```

</head>

<h1>Wrong credentials</h1>

<body>

<p>

access not authorized for host : <%=Error.getError()%><br>

msg : <%=Error.getMessage()%><br>

<a href="index.html" return to the index></a>

</p>

```

Et également le code de la page maJspLogge.jsp :

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<%@ page import="com.example.javatt06.Error"%>

<%@ page import="com.example.javatt06.Info"%>

<jsp:useBean id="Info" class="com.example.javatt06.Info" scope="session"/>

<html>

<head>

    <title>Logged</title>

</head>

<body>

<h1>You are logged</h1>

<p>

    Your username : <%=Info.getUsername()%><br>


    Your password : <%=Info.getPassword()%><br>

</p>

```

Voici le résultat que ça a donné si on s'est connecté avec les bons credentials :

Page de login

Votre nom d'utilisateur :
 Votre mot de passe : 

You are logged

Votre nom est admin
 Votre password est Pa\$\$w0rd

Exercice 7

On récupère la page HTML de l'exercice 3 et on change le nom par « Servlet » :

```
<form method="post" action="Servlet">
```

Voici le code Servlet :

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet Servlet at " + request.getContextPath() + "</h1>");
        out.println("<p>Username = " + request.getParameter("username") + "</p>");
        out.println("<p>Password = " + request.getParameter("password") + "</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

On crée une page en HTML avec une classe Servlet et on affiche « Servlet Servlet At » et je vais également afficher les IDs.

Voici le résultat :

Servlet MaServlet at /ServletSimple

Votre nom d'utilisateur est : admin et votre mot de passe est : Pa\$\$s0rd

Exercice 8

On reprend les différents Beans et on crée une classe Servlet. Dans Servlet, on implémente la méthode « doPost ». Il faut ensuite créer une session, on récupère le username et password et on vérifie s'ils sont corrects. Si c'est bon on met les nouveaux identifiants.

Voici le code de la classe GatewayServlet :

```

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    HttpSession session = request.getSession();

    session.setMaxInactiveInterval(20);

    String username = request.getParameter("username");

    String password = request.getParameter("password");

    if(username.equals("admin") && password.equals("emf123")){

        BeanInfo inf = new BeanInfo();

        inf.setNom(username);

        session.setAttribute("Info", inf);

        RequestDispatcher rd = request.getRequestDispatcher("maJspLogge.jsp");

        rd.forward(request, response);

    }else{

        BeanError err = new BeanError();

        err.setMessage("Wrong credentials");

        RequestDispatcher rd = request.getRequestDispatcher("erreur.jsp");

        rd.forward(request, response);

    }

}

}

}

```

Code de la page JSP MaJspLogge.jsp :

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ page import="com.example.javatt08.beans.BeanError"%>

<%@ page import="com.example.javatt08.beans.BeanInfo"%>

<jsp:useBean id="Info" class="com.example.javatt08.beans.BeanInfo" scope="session"/>

<html>

<head>

<title>Logged</title>

</head>

<body>

```

```
<h1>You are logged</h1>

<p>

Your username : <%=Info.getNom()%><br>
```

Code de la page JSP erreur.jsp :

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ page import="com.example.javatt08.beans.BeanError"%>
<%@ page import="com.example.javatt08.beans.BeanInfo"%>
<jsp:useBean id="Error" class="com.example.javatt08.beans.BeanError" scope="session"/>
<html>
<head>
<title>Error</title>
</head>
<h1>Wrong credentials</h1>
<body>
```

Résultat:

Vous n'êtes pas logger

Votre Host est 0:0:0:0:0:0:1
Votre message est 0:0:0:0:0:0:1

Vous êtes logger

Votre nom est admin
Votre password est Pa\$\$w0rd

Exercice 10

Client

Pour le client nous allons commencer à faire 2 boutons qui va nous permettre de faire une demande de récupérer l'auteur et le message.

```
<form method="post" action="ServletCtrl">
<input type="submit" name="getMessage" value="Récupérer message"/>
<input type="submit" name="getAuthor" value="Récupérer auteur"/>
</form>
```

Ensuite nous faisons 2 classes dont une classe est un servlet. La premier classe ClientMessage va nous permettre de faire la communication avec le serveur.

```
public class ClientMessage {
private WebTarget webTarget;
```

```

private Client client;

private static final String BASE_URI = "http://leonettim.emf-
informatique.ch/javaSimple_Rest_Server/webresources";

public ClientMessage() {

client = javax.ws.rs.client.ClientBuilder.newClient();

webTarget = client.target(BASE_URI).path("tutoriel");
}

public String getAuthor() throws ClientErrorException {

WebTarget resource = webTarget;

resource = resource.path("getAuthor");

return resource.request(javax.ws.rs.core.MediaType.TEXT_PLAIN).get(String.class);

}

public String getMessage() throws ClientErrorException {

WebTarget resource = webTarget;

resource = resource.path("getMessage");

return resource.request(javax.ws.rs.core.MediaType.TEXT_PLAIN).get(String.class);

}

public void close() {

client.close();

}

}

```

La classe ServletCtrl on a dû faire des ifs pour savoir si nous voulions recevoir le message, l'auteur ou s'il y a une erreur.

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try ( PrintWriter out = response.getWriter()) {

/* TODO output your page here. You may use following sample code. */

if (request.getParameter("getMessage")!=null){

String reponse = client.getMessage();

out.println(reponse);

}

}

```

```

else if (request.getParameter("getAuthor") != null){
String reponse = client.getAuthor();
out.println(reponse);
}
else {
out.println("Quelque chose ne s'est pas bien passé !");
}
}
}
}

```

Serveur

Dans le serveur nous avons dû aussi créer deux classes, une qu'on va appeler « ApplicationCon-fig » avec son path « webresources ». Ce path va nous permettre de rediriger vers les bonnes ressources.

```

@javax.ws.rs.ApplicationPath("webresources")
public class ApplicationConfig extends Application {

```

Pour la classe Message nous avons aussi dû mettre un path « tutorial » et ensuite pour chaque méthode nous avons aussi dû mettre un path. Si nous mettons « /webresources/tutorial/ getMes-sage » cela va nous donner « Bonjour tout le monde »

```

@Path("tutorial")
public class Message {
    @Context
    private UriInfo context;
    public Message() {
    }
    @GET
    @Path("getMessage")
    @Produces(javax.ws.rs.core.MediaType.TEXT_PLAIN)
    @Consumes(javax.ws.rs.core.MediaType.APPLICATION_FORM_URL_ENCODED)
    public String getMessage() {
        return "Bonjour tout le monde !";
    }
    @GET
    @Path("getAuthor")

```

```
@Produces(javax.ws.rs.core.MediaType.TEXT_PLAIN)
@Consumes(javax.ws.rs.core.MediaType.APPLICATION_FORM_URLENCODED)
public String getAuthor() {
    return "Fait par Leonetti matteo !";
}
```