

BÀI THỰC HÀNH 01

Đề bài:

Tạo một servlet để khi người dùng truy cập vào **http://localhost/sayhello** sẽ hiển thị dòng thông báo “Hello World”.

Hướng dẫn:

- Tạo một servlet HelloWorld.java
- Định nghĩa ánh xạ servlet (Servlet mapping) trong file web.xml

HelloWorld.java có dạng:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Kế thừa lớp HttpServlet để định nghĩa một servlet.
public class HelloWorld extends HttpServlet {

    // Phương thức init() được gọi một lần duy nhất,
    // khi web server load servlet lên lần đầu (để khởi tạo các giá trị cho servlet).
    public void init() throws ServletException {
        // Do required initialization
    }

    // Các phương thức doGet(___), doPost(___) sẽ được gọi mỗi khi có request từ client
    // tùy thuộc vào kiểu request (method = POST hay method = GET)
    // mà phương thức doPost hay doGet sẽ được gọi.

    // Lưu ý rằng: mỗi lần có request,
    // web server sẽ tạo một luồng (thread) để gọi đến các phương thức này
    // trên cùng một instance của servlet(instance của servlet chỉ tạo một lần duy nhất)

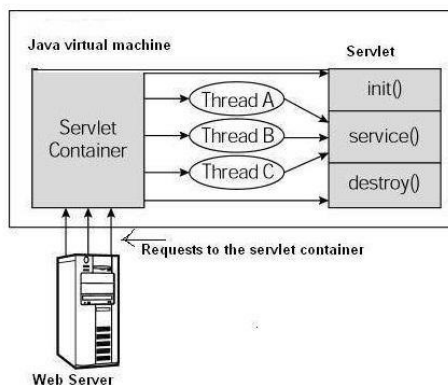
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Định nghĩa response content type
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<h1>" + Hello World + "</h1>");

        out.close();
    }

    public void destroy() {
        // do nothing.
    }
}
```



File web.xml có dạng:

```
<servlet>
  <servlet-name>HelloWorld</servlet-name>
  <servlet-class>HelloWorld</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>HelloWorld</servlet-name>
  <url-pattern>/sayhello</url-pattern>
</servlet-mapping>
```

BÀI THỰC HÀNH 02

Đề bài:

Tạo một trang **login.html**, cho phép nhập **user name**, **password**.

Khi người dùng bấm nút **Login**, sẽ gửi dữ liệu đến servlet **LoginChecker.java**

Servlet **LoginChecker** kiểm tra:

- nếu user name là “admin” và password là “123” thì in ra câu thông báo: “Xin chào”,
- ngược lại, in ra câu thông báo: “Tên đăng nhập hoặc mật khẩu không đúng”

Hướng dẫn:

Trang **login.html**

```
<form action="check" method="get">
  User name: <input type="text" name="user"><br>
  Password: <input type="password" name="pass"><br>
  <input type="submit" value="Login">
</form>
```

File **LoginChecker.java**

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class LoginChecker extends HttpServlet{
public void doGet(HttpServletRequest request,HttpServletResponse response)
throws ServletException,IOException
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    String user = request.getParameter("user");
    String pass = request.getParameter("pass");

    if(user.equals("admin") && pass.equals("123")) {
        Out.println("Xin chào: " + user);
    } else {
        Out.println("Vui lòng đăng nhập lại");
    }
    out.close();
}}
```

File **web.xml** có dạng:

```
<servlet>
  <servlet-name>LoginChecker</servlet-name>
  <servlet-class>LoginChecker</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>LoginChecker</servlet-name>
  <url-pattern>/check</url-pattern>
</servlet-mapping>
```

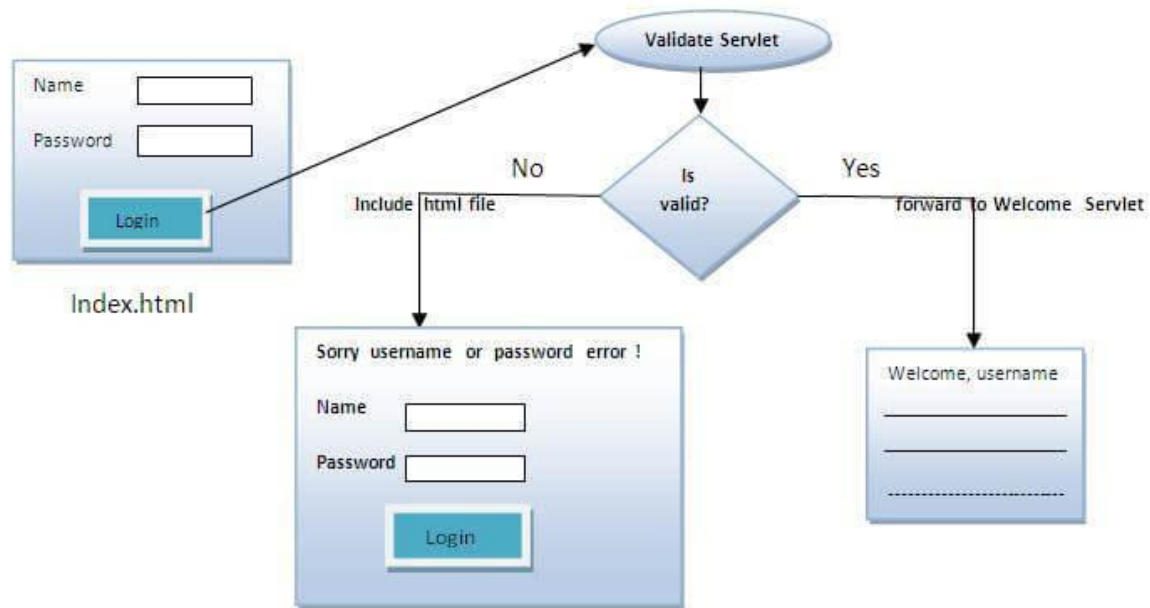
BÀI THỰC HÀNH SỐ 03

Đề bài:

Tạo một trang index.html (cho phép nhập vào Name + Password)

Khi bấm vào nút Login, sẽ submit dữ liệu đến servlet Login.java để kiểm tra

- Nếu Name là admin và Password là 123 → **forward** đến servlet **Welcome.java** (tại servlet Welcome, hiển thị câu: “Welcome” + Name)
- Ngược lại, servlet Login hiển thị câu: “Sorry username or password error!” và **include** form đăng nhập (Login.html) ở ngay bên dưới câu thông báo lỗi.



Hướng dẫn:

Tham khảo code ở trang: <https://www.javatpoint.com/requestdispatcher-in-servlet>

File index.html có dạng:

```
<form action="servlet1" method="post">
  Name:<input type="text" name="userName"/><br/>
  Password:<input type="password" name="userPass"/><br/>
  <input type="submit" value="login"/>
</form>
```

File Login.java có dạng:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Login extends HttpServlet {

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

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String n=request.getParameter("userName");
        String p=request.getParameter("userPass");
```

```

        if(n.equals("admin") && p.equals("123")){
            RequestDispatcher rd=request.getRequestDispatcher("servlet2");
            rd.forward(request, response);
        }
        else{
            out.print("Sorry UserName or Password Error!");
            RequestDispatcher rd=request.getRequestDispatcher("/index.html");
            rd.include(request, response);
        }
    }
}

```

File WelcomeServlet.java có dạng:

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class WelcomeServlet extends HttpServlet {

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

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String n=request.getParameter("userName");
        out.print("Welcome "+n);
    }
}

```

File web.xml có dạng:

```

<web-app>
    <servlet>
        <servlet-name>Login</servlet-name>
        <servlet-class>Login</servlet-class>
    </servlet>

    <servlet>
        <servlet-name>WelcomeServlet</servlet-name>
        <servlet-class>WelcomeServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>Login</servlet-name>
        <url-pattern>/servlet1</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>WelcomeServlet</servlet-name>
        <url-pattern>/servlet2</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>
</web-app>

```

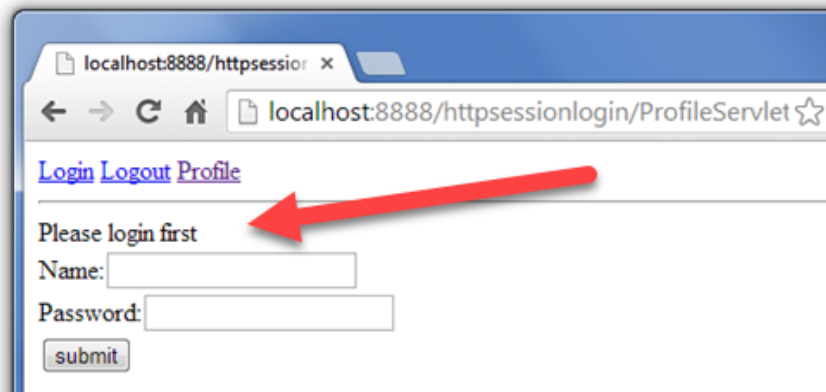
BÀI THỰC HÀNH SỐ 04

Đề bài:

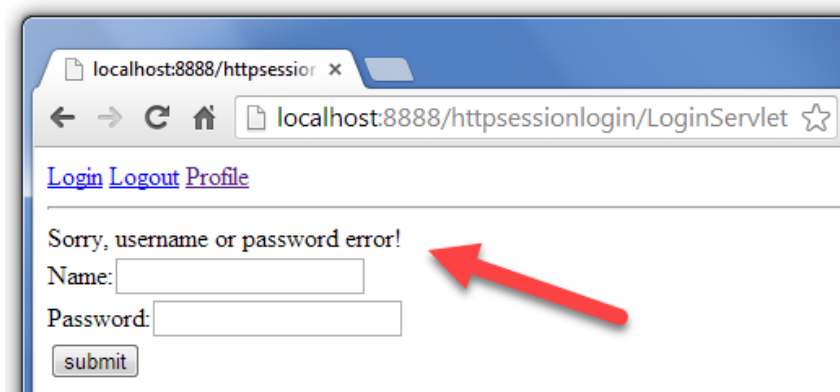
Dùng kỹ thuật Cookie, RequestDispatcher thực hiện yêu cầu sau:

Xây dựng một trang web có 3 liên kết sau: Login, Logout, Profile.

Nếu chưa đăng nhập mà bấm vào liên kết Profile, sẽ hiện câu thông báo **“Please login first”** và đồng thời hiển thị form Đăng nhập ngay bên dưới.

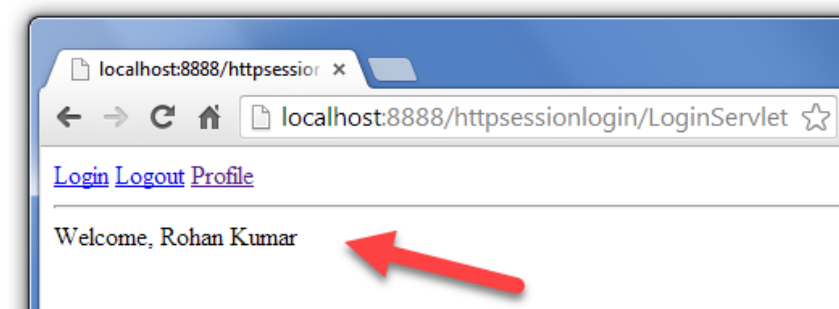


Ở form Đăng nhập, nếu người dùng đăng nhập sai (Giả sử: Đăng nhập đúng là đăng nhập với mật khẩu = 123), sẽ hiển thị thông báo **“Sorry, username or password error!”** và đồng thời hiển thị form Đăng nhập ngay bên dưới.

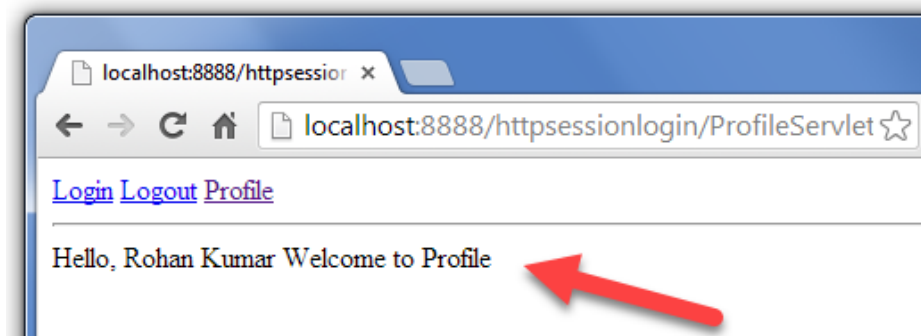


Nếu người dùng đăng nhập đúng (Mật khẩu là 123) thì chuyển đến trang Welcome và hiển thị câu thông báo **“Welcome + tên người đăng nhập”**

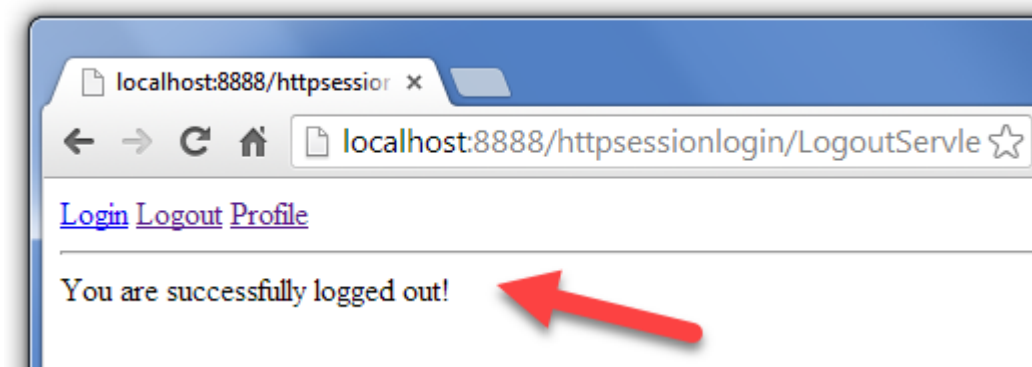
Lưu ý: Tạo Cookie, lưu lại tên người dùng đã đăng nhập.



Khi đã đăng nhập rồi, thì bấm vào liên kết Profile, sẽ hiển thị thông báo **“Hello + <tên người đăng nhập>. Welcome to profile”**



Khi người dùng nhấn liên kết Logout, thì xóa **Cookie** và hiển thị thông báo “You are successfully logged out”



Hướng dẫn:

Xem source code: <https://www.javatpoint.com/servlet-login-and-logout-example-using-cookies>

BÀI THỰC HÀNH SỐ 05

Đề bài:

Dùng kỹ thuật Session, RequestDispatcher thực hiện yêu cầu như bài tập số 04.

Hướng dẫn:

Xem source code: <https://www.javatpoint.com/servlet-http-session-login-and-logout-example>

BÀI THỰC HÀNH SỐ 06

Đề bài:

- Thực hiện yêu cầu tương tự như bài tập số 04.
- Ở bài tập này, sinh viên tạo Cơ sở dữ liệu, một bảng **NguoiiDung** có các cột **UserName** (khóa chính) và **Password**.
- Thực hiện kết nối xuống CSDL để kiểm tra đăng nhập.
- Dùng **Session** để lưu khi đăng nhập thành công.
- Xóa **Session** khi người dùng Logout.