

Assignment 5:

Dynamic Web Pages, Part 2: JSP & Java Servlets

- This assignment will be introduced in a class meeting on November 22nd, 2016 at 11:30 in seminar room 0.124.
- The groups will present the solution of this assignment on November 29th, 2016 at 11:30. We will meet in seminar room 0.124.
- You may work on the assignments in groups of up to three people (if possible, keep the same groups as in the previous assignments).
- All members of a group have to show up together for the grading of the assignment. For the grading, each group will need to have the source code ready for the discussion as well as the running implementation to be presented by the group. Moreover, each group member might be asked questions about the solution.
- If you have questions, send an email to adnan.tariq@ipvs.uni-stuttgart.de.

Task 1 – Java Server Pages (JSP)

Write a JSP page that displays the following information:

- The date and time when the page was requested.
- The browser (user agent) that requested the page.
- The IP address and DNS name of the browser that requested the page.

Upload your page to the `public_html` directory in your home directory on the *netappsvm* web server (see below for information about the *netappsvm* web server), and test your page by calling it with a web browser (you might also use different browsers like Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, etc. to check the different output for the user agent).

(Task 2 on next page →)

Task 2 – Java Servlets and Sessions

In this task, you will implement a simple shopping cart using Java servlets and sessions. Basically, the shopping cart consists of two servlets. The first servlet is used to put an item into the shopping cart. This servlet is called from an HTML form that the user uses for specifying the item to be put into the cart. The second servlet displays the current items in the shopping cart.

Task 2.1

Write the HTML form that the user uses to put items into the shopping cart. This form should contain one text field for defining the name of the item to be put into the cart. For simplicity, we assume that the user can use free text for specifying the name of an item. When the user clicks the submit button, the name of the item should be transferred to the servlet implemented in Task 2.2 (see below for information about the URL for calling the servlet).

Upload the form to your `public_html` directory on *netappsvm*.

Task 2.2

Implement the Java servlet that puts an item into the shopping cart. This servlet is called from the web form of Task 2.1 with the name of the item. When the user submits an item to the servlet, the servlet will store the item and return an HTML page with the following text (“X” should be replaced by the current number of items in the cart):

```
New      item      successfully      added      to      shopping      cart.
Your cart contains X items.
```

The content of the shopping cart shall be managed with an HTTP session (Java class `HttpSession`). When the servlet is called for the first time by the user, a new session is created and the first item is stored in the session object. For every subsequent call, a new item is added to the cart.

Compile and upload your servlet to the directory `~/public_html/WEB-INF/classes/` on the *netappsvm* web server (see below for more information about how to deploy and call servlets). Test your servlet with the web form from Task 2.1.

Task 2.3

Write a servlet that displays the content of the shopping cart. This servlet shall read the content of the shopping cart from the `HttpSession` object (see Task 2.2) and return an HTML page with the following information:

```
Item #1:<name of first item>
Item #2:<name of second item>
...
```

Upload your servlet to the directory `~/public_html/WEB-INF/classes/` on the *netappsvm* web server.

Write a simple HTML page to call this servlet and upload it to the *netappsvm* web server. Test the servlet by calling it through this HTML page.

Information

- You can use the web server `netappsvm.informatik.uni-stuttgart.de` to upload and test your HTML pages, JSP pages, and servlets implemented in this assignment. To login to netappsvm, you need an account. If you did not apply for one yet, please contact Adnan Tariq.
- The server *netappsvm* is only accessible from within the University network.
- The servlet engine (Apache Tomcat) for this exercise is running on port 8080 on the server `netappsvm.informatik.uni-stuttgart.de` (see below for example URLs of JSP pages and servlets).
- The web directory is located at `/home/user/public_html/` (replace `user` by your user name). The URL of a file, say `foo.html`, in this directory is

<http://netappsvm.informatik.uni-stuttgart.de:8080/~user/foo.html>

Note the port number 8080 (8080 is the port of the servlet engine that can also serve ordinary HTML pages).

- Similar to HTML pages, JSP pages must be stored in the directory `~/public_html/`. JSP pages must have the extension `".jsp"` to be interpreted by the servlet engine. You must send requests for JSP pages to the servlet engine running on port 8080. That is, the URL of a JSP page, say `foo.jsp`, looks like this:

<http://netappsvm.informatik.uni-stuttgart.de:8080/~user/foo.jsp>

- Compiled servlet classes (extension `".class"`) must be stored in the directory `~/public_html/WEB-INF/classes/` (see below for instructions on how to compile a servlet).

Moreover, you have to create a file `~/public_html/WEB-INF/web.xml` with the following content to tell the servlet engine where to find your servlet implementation (called "MyServlet" in this example):

```
<web-app xmlns=http://java.sun.com/xml/ns/javaee
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation=http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd
version="2.5">

  <servlet>

    <servlet-name>MyServlet</servlet-name>

    <servlet-class>MyServlet</servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>MyServlet</servlet-name>

    <url-pattern>/myServlet</url-pattern>

  </servlet-mapping>

</web-app>
```

Then, you can invoke your servlet using the following URL defined in the `url-pattern` element:

<http://localhost:8080/~user/myServlet>

- To compile servlets, you need the Java Servlet API, which is part of the Java Enterprise Edition (J2EE). These classes are available on the server nimbus in the following file:

```
/usr/share/tomcat6/lib/servlet-api.jar
```

To use these classes, add this jar to your CLASSPATH (one line):

```
export CLASSPATH=./usr/share/tomcat6/lib/servlet-api.jar
```

Then you can compile your servlet, say `FooBar.java`, as usual using the standard Java compiler (also installed on *netappsvm*), and copy the resulting `.class` file to the classes directory (see above).

```
javac FooBar.java
```

- HTML/JSP pages and Java servlet classes have to be readable by the web server and servlet engine, respectively. You can make files readable by everyone (including the web server/servlet engine) using the following command:

```
chmod a+r filename
```