Master 2 MIASHS
2018-2019
Jérôme DAVID

TP1 Java EE

Servlet Basis

Servlets are the “simplest” (or lowest level) web component of JavaEE. This first TP will focus on the main functionalities of the servlets.

1- Preliminaries

Create a new Maven web Application project called “`FirstServlets`” (group id: `fr.uga.miashs`)

2- A first servlet

Create a new servlet called `MyFirstServlet` in the package `fr.uga.miashs.firstservlets`. Run the servlet (right click on the servlet class, then “run”).

Have a look to the method `processRequest` and answer the following questions:

- What represent the two parameters of the method?
- What represent the variable `out`?
- What means the line “`response.setContentType("text/html;charset=UTF-8");`”? What it does (use the developer tools of the browser to display the HTTP response)?

Using the methods of `HttpServletRequest`, rebuilt the query sent by the browser to the server.

Observe the annotation:

```
@WebServlet(name = "MyFirstServlet", urlPatterns = {"/MyFirstServlet"})
```

Add the url pattern `/blabla/toto.html` and observe that now there two ways to access the same servlet:

```
@WebServlet(name = "MyFirstServlet", urlPatterns = {"/MyFirstServlet",
"/blabla/toto.html"})
```

Now add the pattern `/bidule/*` and verify that any url starting with “`/bidule/`“ reach the servlet. Modify your servlet in order to display the context path, the servlet path, the path info and the request uri. Whats is the relation between them?

3- Servlet Lifecycle

Create a Servlet that displays on the browser the number of its instances. Access it several times. What do you observe?

Override the methods `init(ServletConfig config)` and `destroy()`, in order to understand when they are called (i.e. by displaying a message in the logs)..

4- Contexts

When building web application, we usually need to store information into main memory between user requests (I do not speak about database). In a servlet container (i.e. the servlet execution engine), there are usually 3 contexts in which data can be stored:

- the *application context*: data in this scope are shared by the whole app and can last until the end of the app;
- the *session context*: the data in this scope are valid only the context of the session (i.e. the experience of one particular user during some time);
- the r*equest context*: the data stored in this scope are valid only during the processing of the request.

From the body of a servlet class, the different contexts are represented by the following classes:

- Application context: `ServletContext` (obtained by calling: `this.getServletContext()`)
- Session context: `HttpSession` (obtained by calling: `request.getSession()`)
- Request context: `HttpServletRequest` (parameter `request`)

With these objects, you can store objects within a key-value store:

- to store `ctx.setAttribute(String name, Object value)`
- to retrieve: `ctx.getAttribute(String name) : Object`
- to list the attribute names: `ctx.getAttributeNames() :  Enumeration<String>`

Write a servlet that increments, stores and displays two integers each time it is called. The first integer will be stored in the session context, the second one in the application context. Launch the servlet in parallel with two browsers. What do you observe?