

Object Oriented Programming II (Java II)

420-P33-SU

Servlets

Course Objectives

- ♦ Understand what a Servlet is.
- ♦ Understand the main Servlet methods.
- ♦ Understand the life cycle of a Servlet.
- ♦ HttpServletRequest, & HttpServletResponse
- ♦ Write HTML which uses a Servlet
- ♦ Redirection

The HttpServlet Class

Implements the Servlet interface and inherits the following 5 methods:

- init (ServletConfig)
- destroy ()
- service (ServletRequest req, ServletResponse res):
- getServletConfig ()
- getServletInfo ()



allow you to retrieve information about the servlet.

A Servlet

```
@WebServlet(name = "myServlet", urlPatterns = { "/myServlet" }) public
class MyServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

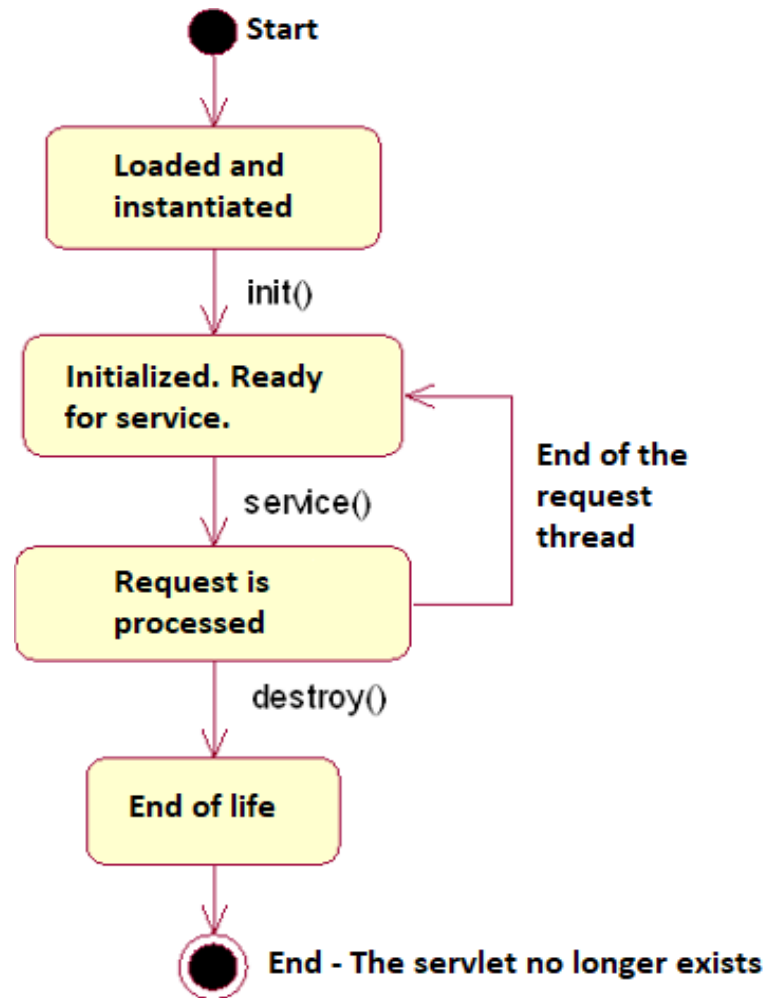
    public MyServlet() {
        super();
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
    ServletException, IOException {
        // TODO Auto-generated method stub
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    {
        // TODO Auto-generated method stub
    }
}
```

A Servlet's Main Methods



- `init()` is called after the Servlet is instantiated
- `service()` is invoked for each request made to the servlet
 - Calls the methods `doGet()`, `doPost()`, `doPut()`, `doDelete()`
- `Destroy()` is called through the application server when the servlet is no longer needed.

The life cycle of a Servlet



HttpServletRequest (interface)



Provides access to the contents of a query using these methods.

- Accessing the parameters of a request
- Access to elements of the HTTP header
- Access to elements in the URL
- Access to client settings
- Access to the session, context and information initialization
- Redirection of the request

Request Parameters

getParameterNames()*: returns Enumeration of parameter names

getParameter(String)*: returns the value of the parameter whose name has been given, if it exists.

getParameterValues(String)*: returns array of String values associated with the parameter whose name has been given.

getParameterMap(): returns a Map object whose keys are the names of the parameters, and the values of the associated parameters.

Parameter values are always of type String, simply because this is the type used in the HTTP header.

Elements in the HTTP Header

getHeaderNames(): returns the names of the header parameters declared in this query as an Enumeration.

getHeader(String), ***getIntHeader(String)*** and ***getDateHeader(String)*** : return the value of the parameter of the header whose name is given, in the form of a String, an int or a Date respectively.

getHeaders() : returns the set of names of the elements of the HTTP header available in this query, in the form of an Enumeration.

getMethod() *: returns the name of the HTTP method used for this request (GET, POST, etc ...).

getContentType() * ***and getContentLength()***: *allow access to the MIME type of the content of the request, and the number of bytes that compose it.*

GetInputStream(): *returns a ServletInputStream bit stream on the request directly, which can be processed without going through the above methods.*

Accessing a Session or Cookie

- ♦ ***getSession()****: returns an object of type HttpSession.
- ♦ ***getCookies()***: returns an array of cookies on the client machine related to the domain name.

HttpServletResponse (interface)



Used to control the response sent to the client

- **Output buffer control**
- **HTTP response control**

Output buffer control

- *setContentType(String)** and *getContentType()*: control the MIME type of the content passed by this buffer. This type must be set before the buffer is sent to the client, even partially.
- methods *getWriter()*** and *getOutputStream()*: allows access to the contents of the response through an object of type `PrintWriter` (extension of `Writer`), or a `ServletOutputStream` object (extension `OutputStream`).

Http Response Control

- HttpServletResponse
 - addHeader (String, String) *,
 - addIntHeader (String, int),
- redirecting the request to another URL:
sendRedirect (String) *

Redirection from a Servlet

- ◆ **requestDispatcher - forward ()** method
redirects by preserving the former request
- ◆ **response.sendRedirect ('My links');**
redirects as if it were a new request

Write HTML into the response

```
Print Writer out = response.getWriter();
//to force a refresh
response.setIntHeader("refresh", 3);
// to write in the response
out.println(
"< html> "+
"< head> "+
"< title> Test < /title> "+
"< /head> "+
"< body> "+
// my html body
"< /body> "+
"< /html> "
);
//close the printer
out.close();
```