RQs + PEs:
Core Servlets

Advanced Topics in Java

Khalid Azim Mughal
khalid@ii.uib.no
http://www.ii.uib.no/~khalid/atij/

Version date: 2006-09-04

REVIEW QUESTIONS - Core Servlets

1. Explain the web application model.
2. Explain the relationship between a servlet, its servlet container and the Servlet API.
3. Which two packages implement the Servlet API?
4. Draw a class diagram to show the relationships between the major classes in the Servlet API, and give a short description of the functionality of these classes.
5. What are the advantages of using servlets to extend server functionality?
6. Describe the request handling cycle, and the methods invoked on the servlet by the servlet container.
7. Explain servlet states and the transitions between them.
8. Why is it recommended for a servlet to override one or more of the `do`*HttpRequestMethodName*`()` methods of the `HttpServlet` class?
9. Who calls the `do`*HttpRequestMethodName*`()` methods?
10. What are the parameters of the `do`*HttpRequestMethodName*`()` methods and who creates them?
11. Explain what is meant by requested URL, requested URI, context path, servlet path, path info and query?

12. Which class provides the methods for reading the parameters submitted by a client?
13. Which class provides the methods for reading the parameters specified in the deployment descriptor file?
14. Which streams are used for sending the contents of the entity-body in the response?
15. What is meant by committing the response?
16. What is meant by document root?
17. Explain the deployment structure for a web application.
18. Outline the main steps in implementing a servlet.
19. Explain the semantics of the HTML `FORM` element.
20. Explain the main differences between `GET` and `POST` methods.
21. What are virtual and real paths, and which method can you use to construct a real path from a virtual path?
22. How are initialization parameters of a servlet specified?
23. Which statements are true?
    - A web application can have more than one `web.xml` file.
    - The `HttpServlet` class implements the `ServletConfig` interface.
    - The `HttpServlet` class implements the `ServletContext` interface.

- In a `do`*HttpRequestMethodName*`()` method, the request and the response objects can be manipulated in any order.
- The servlet configuration can change from request to request.
- The servlet context can change from request to request.
- The `servlet` element must occur before the `servlet-mapping` element in a `web.xml` file.
- The `servlet-class` element must specify the full package name of the servlet class.

PROGRAMMING EXERCISES - Core Servlets

1. Implement a servlet which lists files and directories under the *document root* in which it is installed.
   - Assume that the following *servlet mapping* (in the deployment descriptor file `web.xml`) is defined for this servlet:

     ```
     <servlet-mapping>
         <servlet-name>DirNavigator</servlet-name>
         <url-pattern>/DirNavigator/*</url-pattern>
     </servlet-mapping>
     ```

     Note that the *url pattern* (in the deployment descriptor file `web.xml`) has the suffix `"/*"` which will allow the *path info* for a file or directory to be specified in the request URL.
   - Convert the path info in the URL to a *real path* in order to list the file or directory (See the method `HttpServletRequest.getPathTranslated()`).
   - Some of the methods from the source file `HTTPServer.java` can be adapted for the navigator servlet. However, constructing path names is greatly simplified if you use the approach recommended above.
2. Implement a web application for an order form. The form lists a finite number of items, quantities in stock for each item and the price of each item. The form should allow the user to select items and specify the required quantity of each item. The application should present an invoice to the user for the items ordered.

- In this version all the information is hardwired in a HTML form, and submitted to the servlet which generates the invoice.

3. Implement a web application for FAQs (*Frequently Asked Questions*).
   - The application uses a HTML form to present a list of predefined topic categories.
   - The client can select a topic category to retrieve the FAQs for that particular category, which is generated by a servlet.

   The HTML form passes a unique integer value (`topicID`) to the servlet. This ID identifies a particular topic category.

   The FAQ data is given in a file: `faq.data`. A record in the `faq.data` file consists of three fields: a question (`question`), its answer (`answer`) and the topic ID (`topicID`) of it's category. Three lines specify a single record, for example:

   ```
   What does the status code 400 signify?
   The status code 400 (OK) signifies that the request succeeded normally.
   1001
   ```

   The servlet should read the FAQ data prior to handling the client requests.

   Create a file with suitable FAQ data.