الجامعة السعودية الالكترونية
Saudi Electronic University
2011 1432

Course Title: Web Tech

Course Code : IT 230

Assignment 4

Q1 Write the difference between Buffered Versus Unbuffered Queries. . [1.0 Point]

Answer:

Buffered queries will retrieve the query results and store them in memory on the client side, and subsequent calls to get rows will simply spool through local memory. Buffered queries have the advantage that you can seek in them, which means that you can move the "current row" pointer around in the result set freely because it is all in the client. Their disadvantage is that extra memory is required to store the result set, which could be very large, and that the PHP function used to run the query does not return until all the results have been retrieved.

Unbuffered queries , on the other hand, limit you to a strict sequential access of the results but do not require any extra memory for storing the entire result set. You can start fetching and processing or displaying rows as soon as the MySQL server starts returning them. When using an unbuffered result set, you have to retrieve all rows with mysqli_fetch_row or close the result set with mysqli_free_result before sending any other command to the server.

Q2 Define Cookies and give an example as a coding (servlet) for a cookies? [1.0 Points]

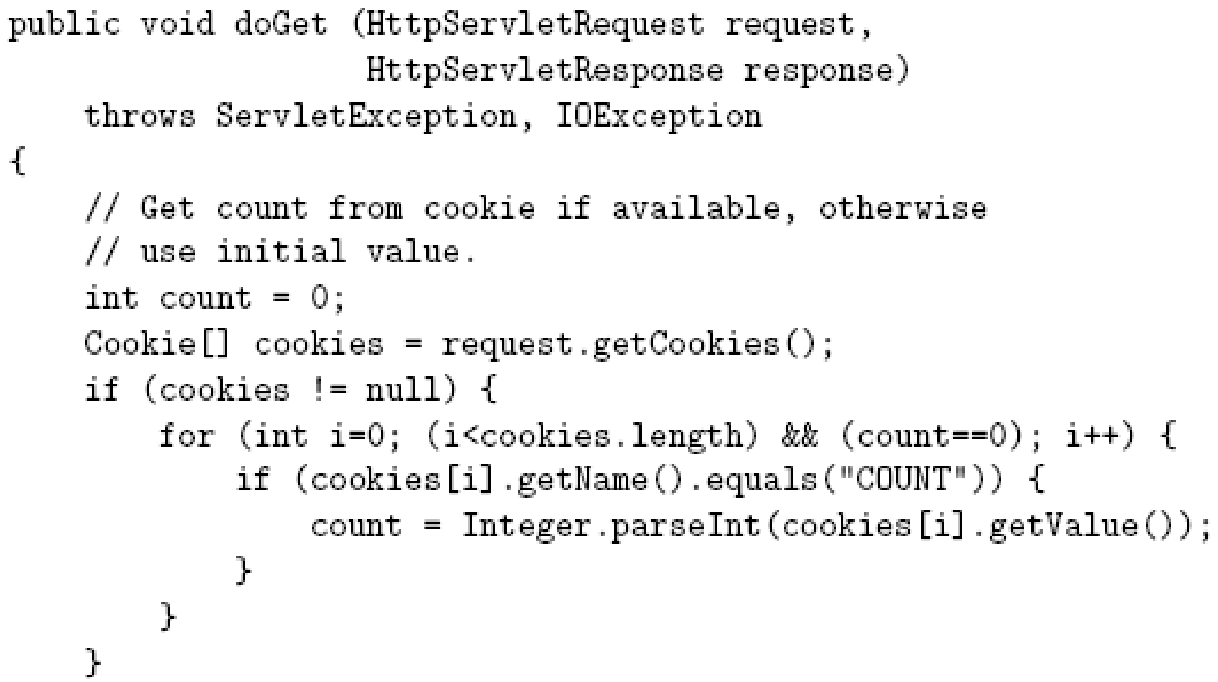
Answer:

A cookie is a name/value pair in the Set-Cookie header field of an HTTP response

Assignment 4

Due date:

`



```
public void doGet (HttpServletRequest request,
                   HttpServletResponse response)
    throws ServletException, IOException
{
    // Get count from cookie if available, otherwise
    // use initial value.
    int count = 0;
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
        for (int i=0; (i<cookies.length) && (count==0); i++) {
            if (cookies[i].getName().equals("COUNT")) {
                count = Integer.parseInt(cookies[i].getValue());
            }
        }
    }
```

Q3: List what the XML document consists of, then select which of the following is a "well formed" XML document: **[1.0 Points]**

1.
   ```
   <?xml version="1.0"?>
   <note>
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
   </note>
   ```

2.
   ```
   <?xml version="1.0"?>
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
   ```

3.
   ```
   <?xml version="1.0"?>
   <note>
   <to age="29">Tove</to>
   <from>Jani</from>
   </note>
   ```

4.
   ```
    <?xml version="1.0"?>
   <note>
   <to age=29>Tove</to>
   <from>Jani</from>
   </note>
   ```

Answer: 1 & 3 are "well formed" XML documents.
An XML document consists of

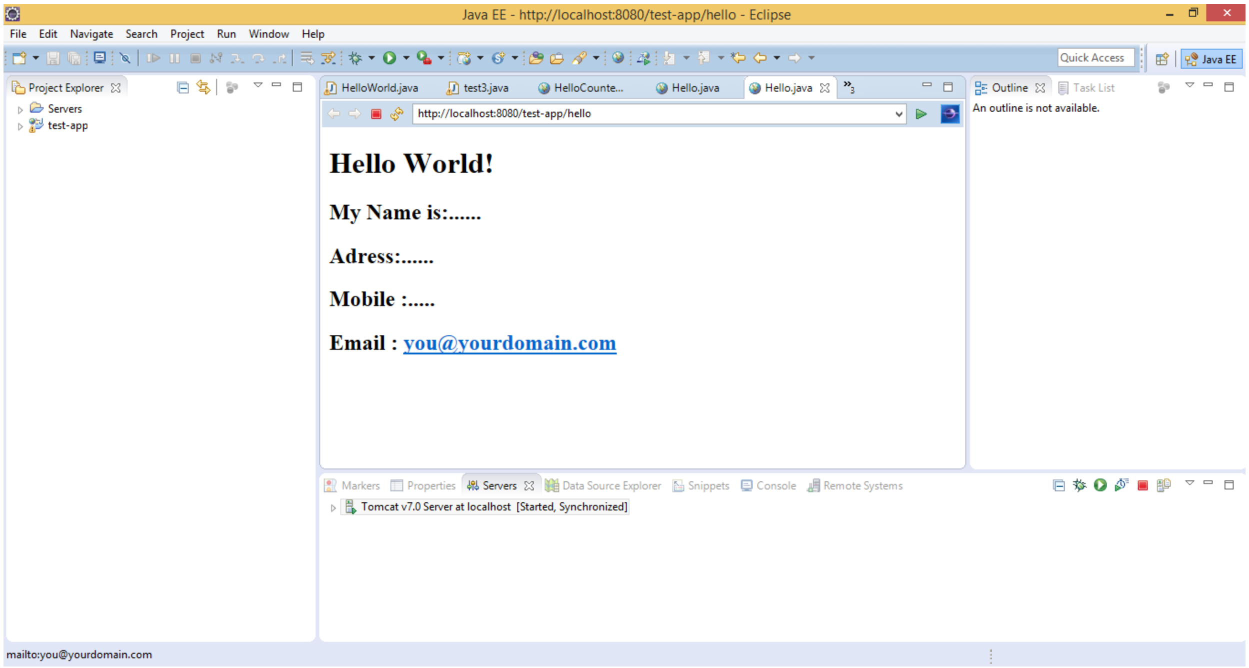
- Markup
  - Tags, which begin with < and end with >
  - References, which begin with & and end with ;
    - Character, e.g. &#x20;
    - Entity, e.g. &lt;
- Character data: everything not markup

- Comments
    - Begin with <!--
    - End -->
    - Must not contain –
- CDATA section
- Special element the entire content of which is interpreted as character data, even if it appears to be markup
    - Begins with <![CDATA[
    - Ends with]]>

Assignment 4

Due date:

Q4: Write a simple servlet that displays the following: [1.0 Points]



http://localhost:8080/test-app/hello

Hello World!

My Name is:......

Adress:......

Mobile :.....

Email : you@yourdomain.com

Tomcat v7.0 Server at localhost [Started, Synchronized]

mailto:you@yourdomain.com

Answer:

```
public class HelloWorld extends HttpServlet {
  @Override
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println
    ("<!DOCTYPE html \n" +
            "    PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \n" +
            "    \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\"> \n" +
            "<html xmlns='http://www.w3.org/1999/xhtml'> \n" +
            "  <head> \n" +
            "    <title> \n" +
            "      Hello.java \n" +
            "    </title> \n" +
            "  </head> \n" +
            "  <body> \n" +
            "     <h1>  Hello World! </h1>" +
            "    <h2> \n" +
            "       My Name is:...... \n" +
            "    </h2> \n" +
            "    <h2> Adress:...... </h2> "+
            "    <h2> Mobile :..... </h2> " +
            "    <h2> Email : <a
href='mailto:you@yourdomain.com'>you@yourdomain.com</a></h2>"+
            "  </body> \n" +
            "</html> ");
  }
}
```