State Management Quiz

1. Can post requests be bookmarked? What are the problems?

POST requests can be bookmarked, but since a bookmark only consists of the URL, all of the form parameters will be lost. This will often mean that the web server doesn't know what to do with the request, since it was expecting some form parameters.

1. What is the purpose of request dispatching?

Servlets may internally pass (“forward”) the request processing to another local resource or to tell the client browser to issue another HTTP request to a specified URL

1. What is the difference between redirect and request dispatch?

|  |  |
| --- | --- |
| **request dispatch (forward)** | **Redirect** |
| * passes the request to another resource on the server   + sometimes referred as “server side redirect“ * request and response objects passed to destination servlet. * Browser is completely unaware of servlet forward and hence the URL in browser address bar will remain unchanged | • server sends HTTP status code 3xx to client along with the redirect URL (usually 302 temporary redirect)  • client then sends a new request to the URL  • extra round trip  • address bar will change to new URL  • only http message sent, request and response objects cannot be sent |

1. What is an attribute ?

* An object bound into one of the three servlet API objects
  + HttpServletRequest
  + HttpSession
  + ServletContext
* Is a name value pair
  + value has type Object
  + name is String

1. What is the difference between attributes and parameters?

|  |  |
| --- | --- |
| **parameters** | **attributes** |
| HTTP request messages send input parameters as name/value pairs. Input parameters are text that must be accessed and converted by a servlet. This is the main mechanism web apps use to send information from the browser to the server | Attributes are objects on the server. They promote  communication between components. Only request attributes are thread-safe. |
| * Request parameters are the result of submitting an HTTP request with a query string that specifies the name/value pairs, or of submitting an HTML form that specifies the name/value pairs. The name and the values are always strings. * Parameters are Strings, and generally can be retrieved, but not set. | * Request attributes (more correctly called "request-scoped variables") are objects of any type that are explicitly placed on the request object via a call to the setAttribute() method. They are retrieved in Java code via the getAttribute() method and in JSP pages with Expression Language references. Always use request.getAttribute() to get an object added to the request scope on the serverside i.e. using request.setAttribute(). * Attributes are objects, and can be placed in the request, session, or context objects. Because they can be any object, not just a String, they are much more flexible. You can also set attributes programmatically and retrieve them later. |

1. What are dangers of using attributes?

Caution when using Context scope attributes, because

* Shared by every servlet and every request in the application
* Like nuclear power
  + very powerful
  + have to be careful
* Not thread safe
  + Nor session attributes
  + Only request attributes thread safe

1. What does it mean to say that http is stateless? Give an example of a stateful protocol.

* HTTP is a stateless protocol, which means that after a web server responds to a request, there is no “memory” of the request or who sent it.
* The File Transfer Protocol (FTP) protocol is stateful: the client establishes a Control Connection for the duration of an FTP session that typically spans multiple data transfers.

1. Give 5 different methods for maintaining state information (count each attribute scope as one method).

|  |  |
| --- | --- |
| **State** | **Method** |
| request scope: destroyed when servlet finishes processing request | request.setAttribute("requestState", "bob"); |
| session scope: destroyed when user closes browser | request.getSession().setAttribute("sessionState", "bob"); |
| application scope destroyed when Container stopped. | this.getServletContext().setAttribute("applicationState", "bob"); |
| Cookies saved on browser,  temporary (deleted when the browser closes)  permanent | Cookie[] cookies = request.getCookies(); |
| Hidden fields on a form | request.getParameter("hiddenState")); |

1. How long does a session last?

It depends on

* session timeout in the DD

<session-config>

<session-timeout>30 //30 minutes

</session-timeout>

</session-config>

</web-app>

* session.setMaxInactiveInterval(20\*60); //seconds
* session.invalidate(); //remove session immediate

1. What is a cookie, and how long does a cookie last?

* An HTTP cookie (also called web cookie, Internet cookie, browser cookie, or simply cookie) is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing.
* temporary cookie
  + browser removes when it closes
  + this is default
  + session cookies are like this
* permanent cookie
  + a cookie that has a max age set
* If the browser is configured to store cookies, it will then keep this information until the expiry date.

1. What is the purpose of URL rewriting?

If your browser does not support cookies, URL rewriting provides you with another session tracking alternative. URL rewriting is a method in which the requested URL is modified to include a session ID

Note: It will work only with links ad can send only text

1. Why does the request attribute report ‘null’ for the maintaining state demo?

**protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

System.***out***.format(**"requestState: %s\r\n"**,  
 request.getAttribute(**"requestState"**));

}

The line of code request.getAttribute(**"requestState"**)); report “null” because each time we pass to the servlet the request attributes are reset.

The setAttribure is called in the doGet request.

**protected void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
  
 request.setAttribute(**"requestState"**, **"bob"**);

}