Srinivas K
Vasu34k@gmail.com

- **What is the difference between JSP and Servlets ?**
  JSP is used mainly for presentation only. A JSP can only be HttpServlet that means the only supported protocol in JSP is HTTP. But a servlet can support any protocol like HTTP, FTP, SMTP etc.

- **What is session?**
  The session is an object used by a servlet to track a user’s interaction with a Web application across multiple HTTP requests. The session is stored on the server.

- **Difference between GET and POST**
  In GET your entire form submission can be encapsulated in one URL, like a hyperlink. query length is limited to 255 characters, not secure, faster, quick and easy. The data is submitted as part of URL.
  In POST data is submitted inside body of the HTTP request. The data is not visible on the URL and it is more secure.

- **What is servlet context ?**
  The servlet context is an object that contains a information about the Web application and container. Using the context, a servlet can log events, obtain URL references to resources, and set and store attributes that other servlets in the context can use.

- **How many JSP scripting elements are there and what are they?**
  There are three scripting language elements: declarations, scriptlets, expressions.

- **How can I implement a thread-safe JSP page?**
  You can make your JSPs thread-safe adding the directive <%@ page isThreadSafe="false" % > within your JSP page.

- **What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher()?**
  In request.getRequestDispatcher(path) in order to create it we need to give the relative path of the resource. But in resourcecontext.getRequestDispatcher(path) in order to create it we need to give the absolute path of the resource.

- **8.What are the lifecycle of JSP?**
  When presented with JSP page the JSP engine does the following 7 phases.
  Page translation: -page is parsed, and a java file which is a servlet is created.
  Page compilation: page is compiled into a class file

Page loading : This class file is loaded.
Create an instance :- Instance of servlet is created
jspInit() method is called
_jspService is called to handle service calls
_jspDestroy is called to destroy it when the servlet is not required.

- **What are context initialization parameters?**
  Context initialization parameters are specified by the in the web.xml file, these are initialization parameter for the whole application.

- **What is a Scriptlet?**
  A scriptlet can contain any number of language statements, variable or expressions that are valid in the page scripting language. Within scriptlet tags, you can declare variables to use later in the file, write expressions valid in the page scripting language, use any of the JSP implicit objects or any object declared with a . Generally a scriptlet can contain any java code that are valid inside a normal java method. This will become the part of generated servlet's service method.

- **What is client side refresh?**
  The standard HTTP protocols ways of refreshing the page, which is normally supported by all browsers.
  <META HTTP-EQUIV="Refresh" CONTENT="5; URL=/servlet/MyServlet/">

  This will refresh the page in the browser automatically and loads the new data every 5 seconds.

- **What is the Max amount of information that can be saved in a Session Object ?**
  There is no such limit on the amount of information that can be saved in a Session Object. The only limit is the Session ID length , which should not exceed more than 4K.

- **What are the differences between a session and a cookie?**
  Session is stored in server but cookie stored in client. Session should work regardless of the settings on the client browser. There is no limit on the amount of data that can be stored on session. But it is limited in cookie. Session can store objects and cookies can store only strings. Cookies are faster than session.

- **What is the difference between an attribute and a parameter?**
  The return type of attribute is object, where the return type of parameter is String. The method to retrieve attribute is getAttribute () where as for parameter is getParameter (). We have a method setAttribute to set an attribute. But there is no setters available for setting a parameter.

- **How can I invoke a JSP error page from a servlet?**
  You can invoke the JSP error page and pass the exception object to it from within a servlet. For that you need to create a request dispatcher for the JSP error page, and pass the exception object as a javax.servlet.jsp.jspException request attribute.

- **How can I enable session tracking for JSP pages if the browser has disabled cookies?**
  By default session tracking uses cookies to associate a session identifier with a user. If the browser does not support cookies, or if cookies are disabled, you can still use session tracking using URL rewriting. For URL rewriting to be effective, you need to append the session ID for each and every link that is part of your servlet response. By using the methods response.encodeURL() and response.encodeRedirectURL() we can achieve this.

- **How can we set the inactivity period on a per-session basis?**
  We can set the session time out programmatically by using the method setMaxInactiveInterval() of HttpSession.

- **What is a Session Id?**
  It is a unique id assigned by the server to the user when a user starts a session.

- **Why we are used setMaxAge() and getMaxAge() in Cookies ?**
  setMaxAge : Sets the maximum age of the cookie. The cookie will expire after that many seconds have passed. Negative values indicate the default behavior, and will be deleted when the browser closes.
  getMaxAge : Returns the maximum specified age of the cookie. If none was specified, a negative value is returned, indicating the default behavior described with setMaxAge.

- **What is the use of setSecure() and getSecure() in Cookies ?**
  setSecure method indicates to the web browser that the cookie should only be sent using a secure protocol (https). getSecure method returns the value of the 'secure' flag.

- **How do you communicate between the servlets?**
  We can communicate between servlets by using RequestDespatcher interface and servlet chaining.

- **What is Servlet chaining?**
  Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request. In servlet chaining, one servlet's output is the input of next servlet. This process

continues until the last servlet is reached. Its output is then sent back to the client. We are achieving Servlet Chaining with the help of RequestDispatcher.

.How would you get the browser to request for an updated page in 10 seconds from the server?
response.**setHeader**(“**Refresh**”, 10);

- **What can you do in your Servlet/JSP code to tell browser not to cache the pages?**
  Cache-Control as shown below:
  response.**setHeader**(“Cache-Control”,“no-cache”); //document should never be cached. HTTP 1.1
  response.setHeader(“Pragma”, “no-cache”); //HTTP 1.0
  response.setDateHeader(“Expires”, 0)

➔ **Which mechanism to choose?**

- HttpSession □ There is no limit on the size of the session data kept.
- □ The performance is good.
- □ This is the preferred way of maintaining state. If we use the HTTP session with the application server’s
- persistence mechanism (server converts the session object into BLOB type and stores it in the
- Database) then the performance will be moderate to poor.
- **Note**: When using HttpSession mechanism you need to take care of the following points:
- □ Remove session explicitly when you no longer require it.
- □ Set the session timeout value.
- □ Your application server may serialize session objects after crossing a certain memory limit. This is
- expensive and affects performance. So decide carefully what you want to store in a session.
- Hidden fields □ There is no limit on size of the session data.
- □ May expose sensitive or private information to others (So not good for sensitive information).
- □ The performance is moderate.
- URL rewriting □ There is a limit on the size of the session data.
- □ Should not be used for sensitive or private information.
- □ The performance is moderate.
- Cookies □ There is a limit for cookie size.
- □ The browser may turn off cookies.

- □ The performance is moderate.
- The benefit of the cookies is that state information can be stored regardless of which server the client talks to
- and even if all servers go down. Also, if required, state information can be retained across sessions

    - **How does an HTTP Servlet handle client requests?** All client requests are handled through the **service**()
- method. The service method dispatches the request to an appropriate method like **doGet**(), **doPost**() etc to handle that request.

- What is pre-initialization of a Servlet? **LF**
  By default the container does not initialize the servlets as soon as it starts up. It initializes a servlet when it
- receives a request for the first time for that servlet. This is called **lazy loading**. The servlet deployment descriptor
- (web.xml) defines the <load-on-startup> element, which can be configured to make the servlet container load and
- initialize the servlet as soon as it starts up. The process of loading a servlet before any request comes in is called
- **pre-loading** or **pre-initializing** a servlet. We can also specify the order in which the servlets are initialized.
- <load-on-startup>2</load-on-startup>

=>What is a filter, and how does it work?

: A filter dynamically intercepts requests and responses to transform or use the information contained in the

requests or responses but typically do not themselves create responses. Filters can also be used to transform the

response from the Servlet or JSP before sending it back to client. Filters improve reusability by placing recurring

tasks in the filter as a reusable unit.