1. What is the difference of a web server and a web container?

Web Container or Servlet Container or Servlet Engine: is used to manage the components like servlets, JSP. It is a part of the web server. It is used for dynamically generate the web pages on the server side.

Web Server or HTTP Server: a server which is capable of handling HTTP request send by a client and respond back with a HTTP response.

1. What is a servlet?

Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

1. How do web servers and web containers interact with servlets?

Servlets are managed by the container. When your server (Tomcat) receives a request for a servlet, it hands the request to the container (Catalina).

Finds the correct servlet based on the URL. Allocates a thread for the request and pass the request and response objects to the servlet thread. Call the servlet service() method which calls either doGet() or doPost() based on the type of request. When thread is finished, the container converts the response object to an HTTP response and hand it to the web server who sends it back to the client. The container deletes the request and response objects.

1. Who creates request objects?

The container creates the HTTP request and response objects.

1. Who creates response objects?

The container creates the HTTP request and response objects.

1. How do HTTP messages relate to the doGet and doPost methods in a servlet?

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.

1. What are the states in the servlet lifecycle?

1.Do not exists

2.Initialized

1. Who calls init and when?

The Container calls init() on the servlet instance after the servlet instance is created but before the servlet can service any client requests.

1. Who calls service() and when?

The container calls service() method to process a client's request in new thread (for every request). When the first client request comes in, the Container starts a new thread or allocates a thread from the pool, and causes the servlet’s service() method to be invoked.

1. Who calls doGet and when?

The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

1. Which of init, service, and doGet should you override?

Init and doGet are overridable.

**init()** - If you have initialization code (like getting a database connection or registering yourself with other objects), then you’ll override the init() method in your servlet class.

**service()** - You should NOT override the service() method. Your job is to override the doGet() and/or doPost() methods and let the service() implementation from HTTPServlet worry about calling the right one.

**doGet() or doPost() -**Whichever one(s) you override tells the Container what you support. If you don’t override doPost(), for example, then you’re telling the Container that this servlet does not support HTTP POST requests.

1. In what sense are servlets multi-threaded?

Each request runs in a separate thread. The container runs multiple threads to process multiple requests to a single service() call of a servlet.

1. What are the implications of this for servlet instance variables?

All threads share instance variables.

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

POST/redirect/GET pattern allows the result of a POST to be book marked. It prevents “Are you sure you want to resubmit”.

1. What is the purpose of request dispatching (request forwarding)?

To pass the request to another resource on the server - sometimes referred as "server-side redirect".

1. What is the difference between redirect and request forwarding (aka request dispatch)?

Servlets may internally pass (“forward”) the request processing to another local resource (to another servlet or JPS in same website).

**Forward** - 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.

**Redirect -** 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?

It's key/value pair stored into one of three other servlet API objects (scopes): HTTPServletRequest (single request), HTTPSession (group of requests from same user), and ServletContext (app). Where key is string and value is an object.

1. What are scopes of attributes?

Request, Session, or Context.

1. What is the difference between attributes and parameters?

Parameters are string values. Request parameters from user, or initParameters from web.xml.

getInitParameter("key")

getParameter("key");

Attributes are objects. Request, Session, or Context.

setAttribute("key", object)

removeAttribute("key");

1. What are dangers of using attributes?

* Request scope attributes are only available for that request.

Request attributes are thread safe - request.setAttribute("myAttr", test);

- Session scope attributes are collections of objects (attributes) that are unique to a set of connected requests from a single browser.

Not thread safe. (Many Tabs could be opened using same session)

request.getSession().setAttribute("MyAttr", test);

- Context scope attributes are available for the whole application (Shared by every servlet and every request in the application).

Not thread safe. (Many servlets can access same Context)

getServletContext().setAttribute("myAttr", test);

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

HTTP is a stateless protocol (an HTTP request by its own nature does not know anything about previous/other requests). Web applications must therefore explicitly manage state information from one request to the next (like “is this person logged in”, or even “who is this person”).

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

A small amount of information sent by a server to a browser, and then sent back by the browser on future page requests. cookies are used in authentication, user tracking (viewing habits on a particular site) or maintaining user’s preferences. i.e. shopping cart etc.

Cookie's data consists of a single name/value pair, sent in the header of the client's HTTP GET or POST request.

The browser is expected to support 20 cookies for each Web server, 300 cookies total, and may limit cookie size to 4 KB each. **Temporary cookie:** browser removes them when it closes (default). **Permanent cookie:** a cookie that has a max age set, and is stored on the client computer. Browser discards all “temporary” cookies when it closes. Every tab or window of browser will have access to all cookies.

1. How long does a session last?

Sessions are saved on the server. While having the ID saved on the client as a JSESSIONID cookie (implicit). The container saves the session info to some datastore, and then sends an HTTP “cookie” with session id back to the browser: Set-cookie (implicit). Tabs share same session because it’s the browser as a whole that maintains the cookies, not individual tabs.

Sessions can become a memory resource issue. Container can’t tell when browser is finished with session.

1. What is a hidden field and how can it be used to maintain state information?

The principle is to include an HTML form with a hidden field containing a session ID in all the HTML pages sent to the client. This hidden field will be returned back to the server in the request. No cookie needed. The disadvantage of this approach is it requires careful and tedious programming effort, as all the pages have to be dynamically generated to include this hidden field. The advantage is all browser supports HTML form.

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

State information can be stored/maintained in request, session, or context scope, and also as hidden fields or cookies.

1. Suppose you create a web application in Eclipse and then create a servlet with the doGet and doPost, and that you map the url stateDemo to the servlet. Now if you were to run the web app and then enter http://localhost:8080/stateDemo in the browser url bar. What will appear in the browser?

It shows 404 - Not Found page as there is no index.html file is created. If there is index.html which has no content, it shows blank page.