1. What is Servlet?

Servlet is a server side technology which process http request to generate static and

dynamic web pages. In the MVC design pattern servlet acts as controller.

2. Life cycle of Servlet ?

Web container is responsible for managing life cycle of servlet. It

Servlet class is loaded.

Servlet instance is created.

init method is invoked.

service method is invoked.

destroy method is invoked.

in servlet life cycle first 3 phases and destroy phase is only invoked once, but service

method is called for each and every request.

3. What is HttpServlet?

The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead, doTrace etc.

4. What are important methods in servlet?

doPut(), doGet(), doPost(), init(), seivce() and destroy().

5. How do you map the Servlets to request urls in web.xml?

For a Java servlet to be accessible from a browser, you must tell the servlet container what servlets to deploy, and what URL's to map the servlets to. This is done in the web.xml file of your Java web application.

Entries to be done in web.xml for servlet-mapping:

<servlet>

<servlet-name>controlServlet</servlet-name>

<servlet-class>com.jenkov.butterfly.ControlServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>controlServlet</servlet-name>

<url-pattern>index.html</url-pattern>

</servlet-mapping>

When there is a request from a client, servlet container decides to which application it should forward to. Then context path of url is matched for mapping servlets.

6. How can you read data from servlet?

getParameter(): You call request.getParameter() method to get the value of a form parameter.

getParameterValues(): Call this method if the parameter appears more than once and returns multiple values, for example checkbox.

getParameterNames(): Call this method if you want a complete list of all parameters in the current request.

7. forward vs sendRedirect?

1. forward sends the same request and responce to another resource. sendRedirect always sends new request because it uses Url of the browser.

2. forward method works at server side. sendRedirect works at client side.

3. forward works within server only. sendRedirect works within server and outside server.

4. forward () is handled internally by the container whereas sednRedirect() is handled by browser.

7. What is Filter?

A filter is an object that is invoked at the preprocessing and postprocessing of a request.

It is mainly used to perform filtering tasks such as conversion, logging, compression, encryption and decryption, input validation etc. The servlet filter is pluggable, i.e. its entry is defined in the web.xml file, if we remove the entry of filter from the web.xml file, filter will be removed automatically and we don't need to change the servlet. So, maintenance cost will be less.

8. Name some of the servlets filters?

There are various types of filters suggested by the specifications:

Authentication Filters.

Data compression Filters.

Encryption Filters.

Filters that trigger resource access events.

Image Conversion Filters.

Logging and Auditing Filters.

MIME-TYPE Chain Filters.

Tokenizing Filters .

XSL/T Filters That Transform XML Content.

9. Difference between Generic Servlet and HttpServlet?

1. Generic servlet is protocol independent whereas HttpServlet is protocol specific. HttpServlet provides additional functionalities such as state management etc.

2. GenericServlet supports only service () method does not contain doGet() and doPost() methods. HttpServlet support doGet(), doPost(), doHead() , doPut() etc.

10. Get vs Post

Get and Post are methods in http request.

1. In Get limited amount of data is send as data is send in header. Large amount of data can be send in Post because data is send in method.

2. Data is not secured in Get whereas data is secured in Post.

3. Get is Idempotent means second request is send only when first request is completed. Post is not Idempotent.

4. Get is more efficient and used more than Post.

5. GET is the default HTTP method whereas we need to specify method as POST to send request with POST method.

11. What is servlet Collaboration?

When one servlet communicates with another resource like html, jsp or another servlet it is known as servlet collaboration.

The communication can be done using

1. RequestDispatcher interface.

2. sendRedirect() method.

12. Which Http methods are Idempotent?

Http method is said to be Idempotent if it returns same result every time. Get, Put, Delete, Head, Options are Idempotent whereas Post is not Idempotent.

13. What is Servlet Config Object?

ServletConfig object is created by web container for each servlet to pass information to a servlet during initialization. This object can be used to get configuration information from web.xml file. The init parameter in web.xml or WebInitParameter annotation are used to pass servlet config value.

14. What is Servlet Context Object?

javax.servlet.ServletContext interface provides access to web application parameters to the servlet. The ServletContext is unique object and available to all the servlets in the web application. When we want some init parameters to be available to multiple or all of the servlets in the web application, we can use ServletContext object and define parameters in web.xml using <context-param> element. We can get the ServletContext object via the getServletContext() method of ServletConfig.

15. What is Attribute and its scopes?

Servlet attributes are used for inter-servlet communication we can be set, get or removed from one of the following scopes:

1. request scope

2. session scope

3. application scope

The servlet programmer can pass information from one servlet to another using attributes. It is just like passing object from one class to another so that we can reuse the same object again and again.

16. What are common tasks performed by Servlet Container?

Servlet containers are also known as web container, for example Tomcat. Some of the important tasks of servlet container are:

Communication Support: Servlet Container provides easy way of communication between web client (Browsers) and the servlets and JSPs. Container gets request from web client and pass the request to requested servlet and when it gets response it pass the response back to client.

Lifecycle and Resource Management: Servlet Container takes care of managing the life cycle of servlet and jsp's. From the loading of servlets into memory, initializing servlets, invoking servlet methods and to destroy them. Container also provides utility like JNDI for resource pooling and management.

Multithreading Support: Container creates new thread for every request to the servlet and provide them request and response objects to process. So, servlets are not initialized for each request and saves time and memory.

JSP Support: JSPs doesn’t look like normal java classes but every JSP in the application is compiled by container and converted to Servlet and then container manages them like other servlets.

Miscellaneous Task: Servlet container manages the resource pool, perform memory optimizations, execute garbage collector, provides security configurations, support for multiple applications, hot deployment and several other tasks behind the scene that makes a developer life easier.

17. What are init and context parameters?

<init-param> and <context-param> are the static parameters which are stored in web.xml file.

If you want to store particular data which is only confined to a particular servlet scope, then you can use init-param.

The init-param is declared inside the <servlet> tag:

<servlet>

<display-name>HelloWorldServlet</display-name>

<servlet-name>HelloWorldServlet</servlet-name>

<init-param>

<param-name>Greetings</param-name>

<param-value>Hello</param-value>

</init-param>

</servlet>

If you want to store data which is common for whole application and if it doesn't change frequently you can use <context-param>.

Context-param are declared under the tag web-app

<web-app>

<context-param>

<param-name>Country</param-name>

<param-value>India</param-value>

</context-param>

<context-param>

<param-name>Age</param-name>

<param-value>24</param-value>

</context-param>

</web-app>

18. What does load-on-startup do web.xml?

The load-on-startup element of web-app loads the servlet at the time of deployment or server start if value is positive. It is also known as preinitialization of servlet. You can pass positive and negative value for the servlet.

Advantage of load-on-startup element:

As servlet is loaded at first request. That means it consumes more time at first request. If you specify the load-on-startup in web.xml, servlet will be loaded at project deployment time or server start. So, it will take less time for responding to first request.

19. What is Session Tracking and what are Session Tracking Techniques used?

Session simply means a particular interval of time. Session Tracking is a way to maintain state of a user. Http protocol is a stateless protocol. Each time user sends a request to the server, server treats the request as the new request. So, we need to maintain the state of an user to recognize to particular user.

There are four techniques used in Session tracking:

1. Cookies

2. Hidden Form Field

3. URL Rewriting

4. Http Session

20. What are Cookies?

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number. By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet. So, cookie is stored in the cache of the browser. After that if any request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user.

Advantages:

1. Simplest technique of maintaining the state.

2. Cookies are maintained at client side.

Disadvantage of Cookies:

1. It will not work if cookie is disabled from the browser.

2. Only textual information can be set in Cookie object.

21. Different types of Cookies?

There are 2 types of cookies in servlets.

1. Non-persistent cookie: It is valid for single session only. It is removed each time when user closes the browser.

2. Persistent cookie: It is valid for multiple session. It is not removed each time when user closes the browser. It is removed only if user logout or sign-out.

22. URL rewriting?

In URL rewriting, we append a token or identifier to the URL of the next Servlet or the next resource. We can send

parameter name/value pairs using the following format:

url?name1=value1&name2=value2&??

Advantage of URL Rewriting:

1. It will always work whether cookie is disabled or not (browser independent).

2. Extra form submission is not required on each page.

Disadvantage of URL Rewriting:

1. It will work only with links.

2. It can send Only textual information.

23. HttpSession?

When a servlet Container gets request it creates a session id for each user. The container uses this id to identify the particular user. An object of HttpSession can be used to perform two tasks:

1. bind objects

2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

24. How to get the HttpSession object?

The HttpServletRequest interface provides two methods to get the object of HttpSession:

public HttpSession getSession ():Returns the current session associated with this request, or if the request does not have a session, creates one.

public HttpSession getSession(boolean create):Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.