

Servlet interview questions and answers

Q 1. What is servlet?

Servlet is a server side programming language which is used for generating dynamic web pages. It generates web-page as a response of the request received from client(browser).

Q 2. Static webpage vs Dynamic webpage?

The webpages which are same for all the users are static webpages and the webpages that are dynamically generated based on the user's request (that may be different for each user depending on the request) are known as dynamic webpages. Servlet is mainly used for dynamic webpages.

Q 3. Life cycle of a servlet?

Following the the stages of servlet life cycle:

- 1) Loading of Servlet class: The servlet container finds the servlet class mentioned in web.xml file and loads it.
- 2) Servlet instantiation: The object of servlet class gets created in this phase.
- 3) Initialization: Servlet initialization by calling `init()` method.
- 4) Servicing the request: In this phase the servlet service the client request by calling the `service()` method.
- 5) Destroy: Last phase of servlet life cycle. The `destroy()` method free up the servlet instance so that it can be garbage collected.

Q 4. Why we use Servlets?

- 1) To process the input data submitted by user from the screen.
- 2) Generate and return the dynamic response to the user based on the request.
- 3) Ideal programming language for interacting with database based on the user's request.
- 4) A servlet can handle multiple request simultaneously which makes them a perfect choice for the high performing applications.

Q 5. Why Servlet is better than CGI?

- 1) Servlet responses faster than CGI because it uses the multithreading concept to service each request. CGI performance is not that good as it creates a new object for each request while servlet allots a new thread for each request.
- 2) Learning and implementing servlet is quite easier compared to CGI.
- 3) Memory consumption is low in servlet compared to CGI.

Q 6. When Servlet is loaded?

- 1) When servlet container receives the first request from client(browser).

- 2) Admin of the application manually loads the servlet.
- 3) When the webserver(in which the servlet is deployed) gets started.

Q 7. What is Servlet interface and what's the use of it?

Servlet interface is an API for servlets. Every Servlet should either implement the servlet interface or extends the class which already implements the interface. `javax.servlet.GenericServlet` and `javax.servlet.http.HttpServlet` are the Servlet classes that implements Servlet interface, hence every servlet should either implement Servlet interface directly or by extending any of these classes.

Q 8. What is ServletConfig?

`ServletConfig` interface belongs to the package `javax.servlet.ServletConfig`. It is used for passing the configuration parameters to the servlet. Servlet container instantiate it implicitly.

Q 9. What is ServletContext?

Each web application has a common `ServletContext`. All the servlets in the web application can access the `ServletContext`. It has the web-application information & resources which are common and accessible to all the servlets present in the web application.

Q 10. ServletConfig vs ServletContext?

Following are the two main differences between `ServletConfig` and `ServletContext`:

- 1) `ServletConfig` is used for passing the configuration information to the Servlet while `ServletContext` provides set of methods which a Servlet class can use to communicate with Servlet container.
- 2) Each Servlet has a separate `ServletConfig` object while `ServletContext` is common for all the servlets in the web application.
- 3) Parameters of `ServletConfig` are defined under `<init-param>` tags in web.xml file. Parameters of `ServletContext` are defined under `<context-param>` tags in web.xml.

Q 11. Difference between GenericServlet and HttpServlet?

- 1) `GenericServlet` is an abstract class which implements Servlet interface while `HttpServlet` abstract class extends the `GenericServlet` class. In short: `GenericServlet` class is a parent class for `HttpServlet`.
- 2) `GenericServlet` does not support any protocol. `HttpServlet` support HTTP and HTTPS protocol.
- 3) `GenericServlet` cannot handle cookies and session while `HttpServlet` can handle them.

Q 12. Difference between forward() and sendRedirect()?

- 1) In `forward()` the same request is forwarded to the another resource. In `sendRedirect()` new request is send to the redirected resource.

- 2) forward() is taken care by the Servlet container while sendRedirect() is handled by the browser.
- 3) In forward() the URL(uniform resource locator) remains same on web browser. In sendRedirect() the URL changes in the web browser address bar.
- 4) forward() is faster compared to sendRedirect().

Q 13. What is deployment descriptor?

web.xml file of a web application is known as deployment descriptor. It is usually placed inside WEB-INF folder of application. It has the information like Servlet name, Servlet mapping etc. This file tells the Servlet container which Servlet class needs to be called for the given URL pattern.

Q 14. doGet() Vs doPost() methods?

- 1) In doGet(), the parameters are visible in the address bar, they get appended to the URL. In doPost() parameters are not visible in the address bar.
- 2) You can maximum transfer 1024 characters through GET request. doPost() doesn't have any limitations.
- 3) doGet() is not good for sensitive data as the parameters do not get encrypted. In doPost() the parameters are encrypted hence it is more secure compared to doGet().
- 4) Method doGet() allow you to bookmark the resource. doPost() doesn't allow bookmarks.
- 5) doGet() is faster compared to the doPost() method.

Q 15. What is the use of <load-on-startup>?

<load-on-startup> is used for specifying the Servlet files which needs to be loaded during server startup. The servlet files specified in this element are loaded as soon as the server starts, it does not wait for the first request for loading them up. This is how it is specified in web.xml file.

```
<servlet>
  <servlet-name>MyServletNameHere</servlet-name>
  <servlet-class>ServletClassHere-FullyQualified</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

If more than one files are specified then the files will be loaded in the same order in which they have been specified in it.

Q 20. What are the different types of session tracking mechanism supported by Servlets?

- 1) URL rewriting
- 2) Hidden Form Fields
- 3) Cookies
- 4) Secure Socket Layer(SSL) Sessions

Q 21. How URL rewriting maintains session?

In URL rewriting method, the session tracking data has been appended at the end of the URL to track the session.

Q 22. Explain Servlet chaining?

Servlet chaining is a concept where the request is processed in a chain of servlets. First Servlet processes the request partially and passes to the second one, then second servlet process it and passes to third one and so on. The last servlet returns the response to the client (browser).

Q 23. How to invalidate a session in servlet?

By calling `session.invalidate()` method.

Q 24. What are the main functions of Servlet container?

- 1) Servlet life cycle management
- 2) Maintains the interaction between servlet and webserver.
- 3) Providing multithreading support for processing more than one request simultaneously.
- 4) Managing of deployment descriptor web.xml file.

Q 25. What is <session-timeout> ?

The element <session-timeout> is used for specifying the timeout of a Session. This is how it is defined in the web.xml file.

```
<session-config>
    <session-timeout>35</session-timeout>
</session-config>
```

It would set the session timeout to 25 minutes.

Q 26. What is Servlet lazy loading and how it can be avoided?

The Servlet container does not initialize the Servlet on server startup by default. It only initializes a servlet when the it receives the request from the client. This is called lazy loading of Servlet.

By specifying **<load-on-startup>** element for a Servlet we can avoid lazy loading. The servlet files specified in <load-on-startup> are loaded as soon as the web sever starts.

Q 27. Why do we need constructor in servlet even though we have a `init()` method?

init() method is used for initializing the servlet however constructor is required in order to instantiate the Servlet class. Servlet container instantiate the Servlet class.

Q 28. When the Servlet is unloaded?

- 1) Admin manually unloads the servlet.
- 2) Web server shut down.

Q 29. How Servlet maintains session using cookies?

Cookie is a small piece of information, which is sent by a servlet to the Web browser. Cookies gets stored in the browser and returned back to the server when needed. A cookie has a name, a single value, and few other attributes.

Q 30. Why using cookies for session tracking is a bad practice?

There are several disadvantages of using cookies for session tracking. Few of them are:

- 1) Since cookies are stored on client-side (in the client's browser), It will not be available if client browser clears or disables the cookies.
- 2) Implementing cookies for session tracking is much more difficult compared to other session management mechanism.
- 3) Cookies only work for HTTP protocol.

Q 31. How do I get the server info in Servlets?

Use this:

```
getServletContext().getServerInfo()
```

Q 32. How to get the client's IP address in Servlets?

Using this:

```
request.getRemoteAddr()
```

Q 33. Why we use filters in Servlet?

We use filters for:

- 1) Security checks
- 2) Modifying the request or response
- 3) Data compression
- 4) Logging and auditing
- 5) Response compression

Q 34. What all protocols are supported by HTTPServlet?

HTTP and HTTPS protocols.

Q 35. What all protocols are supported by GenericServlet?

GenericServlet abstract class is not specific to any protocol.

Q 36. What are the new features added to Servlet 3?

- 1) Servlet Annotations
- 2) Web Fragments

- 3) Web components addition dynamically
- 4) Asynchronous Processing

Q 37. Do we override service() method?

No, we do not override the **service()** method. We generally override the doPost(), doGet() method based on the requirement.