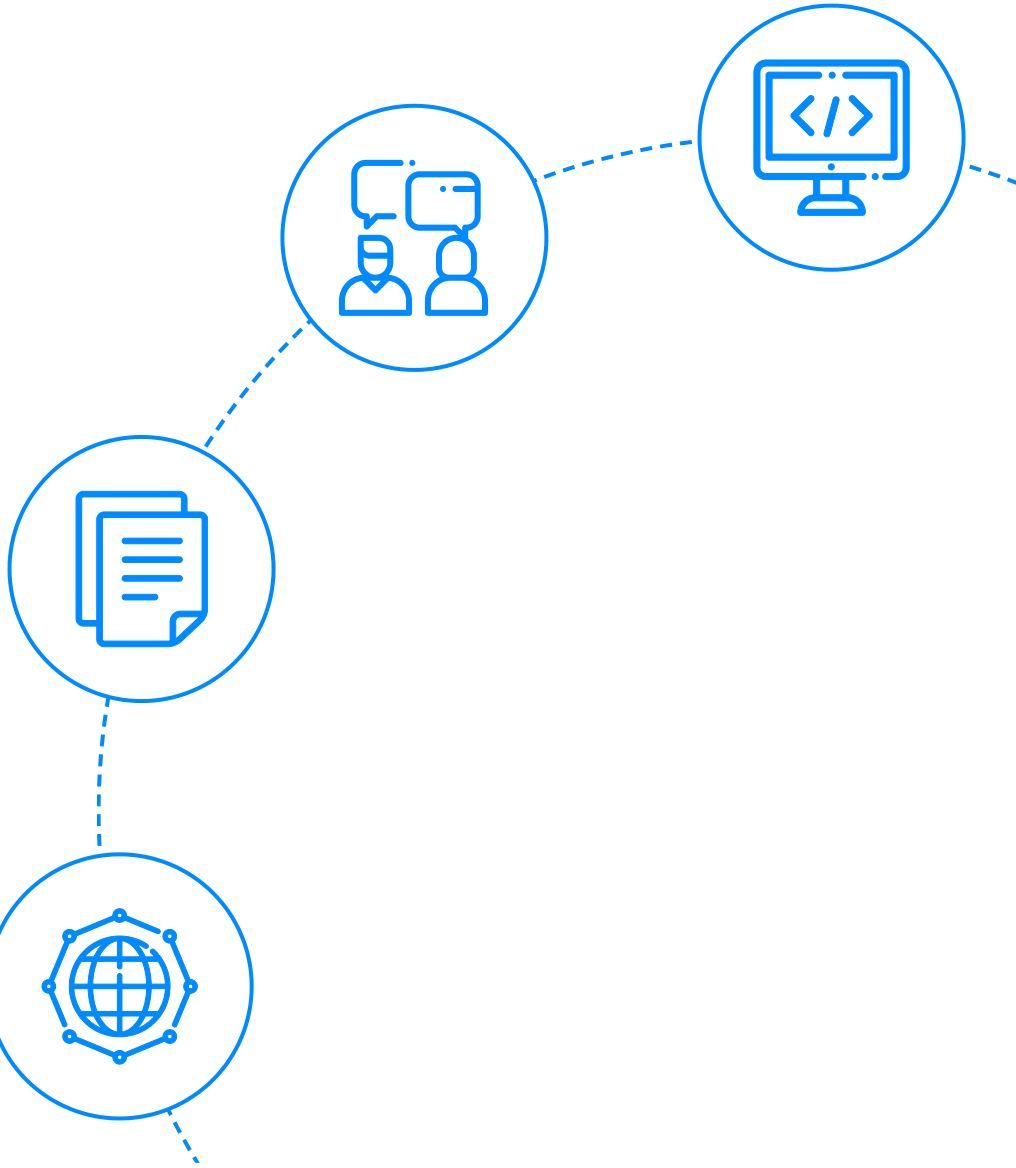
InterviewBit

Servlet Interview Questions



To view the live version of the page, click here.

Contents

Servlet Interview Questions for Freshers

Servlet Interview Questions for Experienced

Servlet Interview Questions for Experienced

(.....Continued)

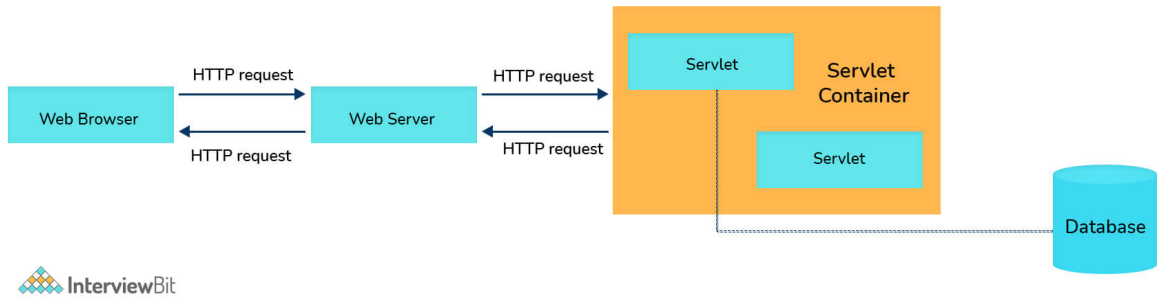
19. What are the methods that a servlet can use to get information about the server?
20. How can a servlet get the name of the server and the port number for a particular request?
21. How can a servlet get information about the client machine?
22. Explain the Single-Thread Model in servlets.
23. How does Background Processing take place in servlets?
24. How does Servlet collaboration take place?
25. Explain Request parameters associated with servlets.
26. What are the three methods of inter-servlet communication?
27. What are the reasons we use inter-servlet communication?
28. What do you mean by Servlet Manipulation?
29. What is the javax.servlet package?

Let's get Started

Introduction to Servlet:

A servlet is an extension to a server. It is a Java class that is loaded to expand the functionality of the server. It helps extend the capability of web servers by providing support for dynamic response and data persistence. These are commonly used with web servers, where they can take the place of CGI scripts. A servlet runs inside a Java Virtual Machine (JVM) on the server, and hence it is safe and portable. Servlets operate only within the domain of the server. These do not require support for Java in the web browser.

The original servlet specification was created by Sun Microsystems. Sun packed Java with Internet functionality and announced the servlet interface. The first version was finalized in June 1997. The servlet specification was developed under the Java Community Process starting with version 2.3. Servlets represent a more efficient architecture as compared to the older CGI.



Servlet Interview Questions for Freshers

1. What is a Servlet?

A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection. Servlets are often used to provide rich interaction functionality within the browser for users (clicking link, form submission, etc.)

2. How do you write a servlet that is part of a web application?

To write a servlet that is part of a web application:
Create a Java class that extends `javax.servlet.http.HttpServlet`.
Import the classes from `servlet.jar` (or `servlet-api.jar`).
These will be needed to compile the servlet.

3. What are some of the advantages of Servlets?

Servlets provide a number of advantages over the other approaches. These include power, integration, efficiency, safety, portability, endurance, elegance, extensibility, and also flexibility. Here are the advantages of servlets:

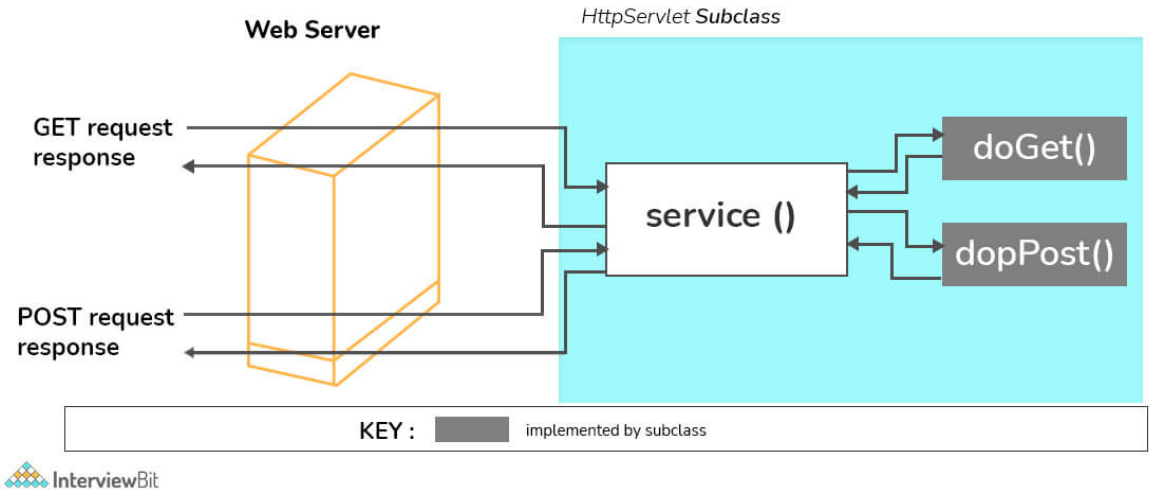
- A Servlet is convenient in modifying regular HTML
- We can write the servlet code into the JSP
- Servlets includes the feature of multithreading of java
- We can make use of exception handling
- Servlets have a separate layer of business logic in the application
- Easy for developers to show and process the information.
- Servlets provide a convenient way to modify HTML pages.
- Servlets have a separate layer of business logic in the application.
- All the advantages of Java-like multi-threading, exception handling, etc. are there in Servlets

4. Explain the Servlet API.

A servlet does not have a main() method, unlike a regular Java program, and just like an applet. It has some methods of a servlet that are called upon by the server for the purpose of handling requests. It invokes the servlet's service() method, every time the server sends a request to a servlet.

To handle requests that are appropriate for the servlet, a typical servlet must override its service() method. The service() method allows 2 parameters: these are the request object and the response object. The request object is used to inform the servlet about the request, whereas the response object is used to then give a response.

As opposed to this, an HTTP servlet typically does not override the service() method. However, it actually overrides the doGet() to handle the GET requests and the doPost() to handle POST requests. Depending on the type of requests it needs to handle, an HTTP servlet can override either or both of these methods.



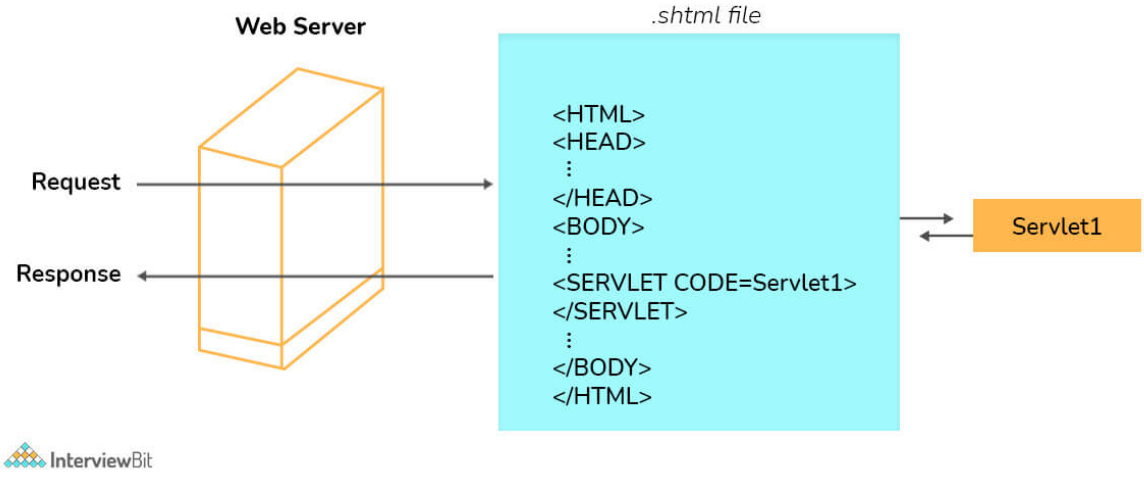
5. What do you mean by server-side include (SSI) functionality in Servlets?

Servlets can be added in HTML pages with the server-side include (SSI) functionality. A page can be preprocessed by the server to add the output from servlets at some points within the page, in the servers that support servlets.

```
<SERVLET CODE=ServletName CODEBASE=http://server:port/dir
          initParam1=initValue1
          initParam2=initValue2>
<PARAM NAME=param1 VALUE=val1>
<PARAM NAME=param2 VALUE=val2>
   Text appearing here indicates that the web server which provides this page does not
</SERVLET>
```

6. Explain the server-side include expansion.

Server-side inclusion (SSI) is a feature of a server in which a placeholder <SERVLET> tag is also returned. The <SERVLET> tag is then substituted by the corresponding servlet code.
The server just parses the pages that are specially tagged, and it doesn't parse and analyses each page it returns. The Java Web Server parses solely pages with a .shtml extension by default. With the SERVLET tag, in contrast to the APPLET tag, the client web browser doesn't see anything between SERVLET and /SERVLET unless SSI is not supported by the server.



7. Define 'init' and 'destroy' methods in servlets.

Servlets Init Method is used to initialise a Servlet.

After the web container loads and instantiates the servlet class and before it delivers requests from clients, the web container initializes the servlet. To customize this process to allow the servlet to read persistent configuration data, initialize resources, and perform any other one-time activities, you override the init method of the Servlet interface.

Example:

```
public class CatalogServlet extends HttpServlet {
private ArticleDBAO articleDB;
public void init() throws ServletException {
articleDB = (ArticleDBAO)getServletContext().
getAttribute("articleDB");
if (articleDB == null) throw new
UnavailableException("Database not loaded");
}
}
```

When a servlet container determines that a servlet should be removed from service (for example, when a container wants to reclaim memory resources or when it is being shut down), the container calls the destroy method of the Servlet interface.

The following destroy method releases the database object created in the init method.

```
public void destroy() {
bookDB = null;
}
```

8. How is retrieving information different in Servlets as compared to CGI?

Servlets have a variety of ways to realize access to information. For the bulk of it, every method returns a specific result. Compared with CGI programs its information by making use of passed environment variables, One can see multiple advantages by using the servlet approach.

- **Stronger type checking:**
  Stronger type checking means that there is more support in the compiler for finding errors in syntax and types. A CGI program utilizes one function to get its environment variables, and several errors are not caught at compile-time and they get only know at runtime cannot be caught until some runtime issue got caused.
- **Delayed calculation:**
  The value for every environment variable has to be precalculated and passed when a server starts a CGI program, even if the program uses it or not. In contrast, servlets launched by servers can enhance the performance on the fly by delaying calculation and do calculations when that piece of code is actually used.
- **Interactives with the server:**
  A CGI program is free from its server, once the execution begins. Then, the single communication path that the program uses is its standard output. However, a servlet can work with the server. A servlet works in 2 ways: either in the server or as a connected sidecar process outside the server.

9. Compare CGI Environment Variables and the Corresponding Servlet Methods.

CGI Environment Variable	HTTP Servlet Method
SERVER_NAME	req.getServerName()
SERVER_SOFTWARE	getServletContext().getServerInfo()
SERVER_PROTOCOL	req.getProtocol()
SERVER_PORT	req.getServerPort()
REQUEST_METHOD	req.getMethod()
PATH_INFO	req.getPathInfo()
PATH_TRANSLATED	req.getPathTranslated()
SCRIPT_NAME	req.getServletPath()
DOCUMENT_ROOT	req.getRealPath("/")
QUERY_STRING	req.getQueryString()
REMOTE_HOST	req.getRemoteHost()
REMOTE_ADDR	req.getRemoteAddr()
AUTH_TYPE	req.getAuthType()
REMOTE_USER	req.getRemoteUser()
CONTENT_TYPE	req.getContentType()
CONTENT_LENGTH	req.getContentLength()
HTTP_ACCEPT	req.getHeader("Accept")
HTTP_USER_AGENT	req.getHeader("User-Agent")

10. How does a servlet get access to its init parameters?

The getInitParameter() method is used by the servlet in order to get access to its init parameters:

```
public String ServletConfig.getInitParameter(String name)
```

The above method returns the value of the named init parameter or if the named init parameter does not exist it will return null. The value returned is always a single string. The servlet then interprets the value.

11. How does a servlet examine all its init parameters?

We can make use of `getInitParameterNames()` function to examine all its init parameters.

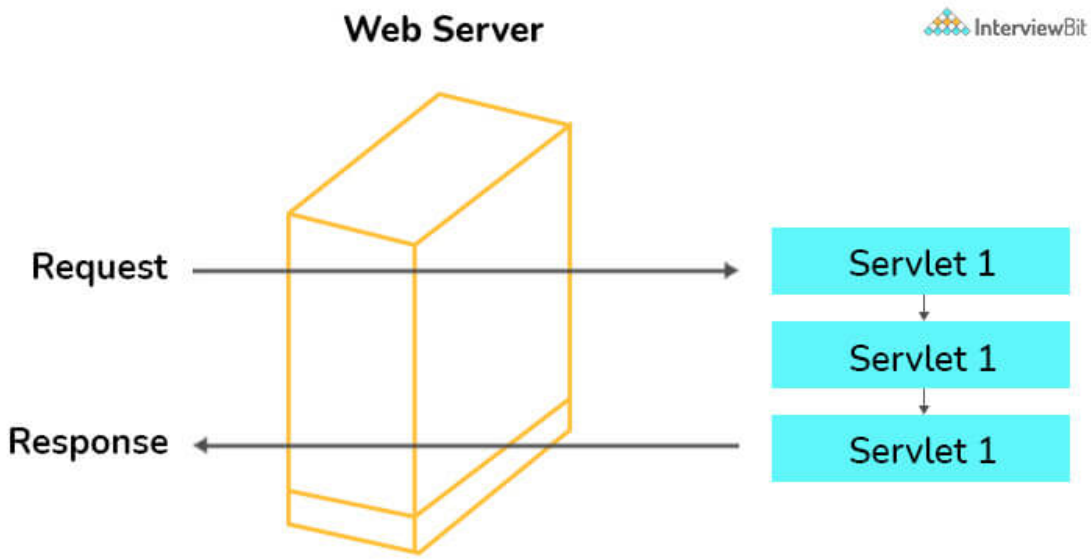
```
public Enumeration ServletConfig.getInitParameterNames()
```

This returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters. This is often used for debugging.

Servlet Interview Questions for Experienced

12. What do you mean by Servlet chaining?

Servlet Chaining is a way where the output of one servlet is piped to the input of another servlet, and the output of that servlet can be piped to the input of yet another servlet and so on. Each servlet in the pipeline can either change or extend the incoming request. The response is returned to the browser from the last servlet within the servlet chain. In the middle, the output out of each servlet is passed as the input to the next servlet, so every servlet within the chain has an option to either change or extend the content. The figure below represents this. Servlets can help in creating content via servlet chaining.



13. What do you mean by ‘filtering’ in servlets?

There are usually 2 ways during which one will trigger a series of servlets for an associate incoming request. In the first manner, it is such that the server that bound URLs ought to be handled with the associated specified chain. the other manner is that one will inform the server to redirect all the output of a selected content through a selected servlet before it's returned to the client. This effectively creates a series on the fly. once a servlet transforms one sort of content into another, this method is named filtering.

14. What are the uses of Servlet chaining?

Given below are some of the use cases of Servlet chaining:

- **Change how a group of pages, a single page, or a type of content appears quickly**

One can talk to those who don’t understand a particular language by dynamically translating the text from the pages to the language that can be read by the client. One can keep away certain words that one doesn’t want others to read.

- **Display in special formats a kernel of content**

For instance, one can add custom tags within a page, and then a servlet can replace these with HTML content.

- **Support for the esoteric data types**

For instance, one can provide a filter that converts nonstandard image types to GIF or JPEG for the unsupported image types.

15. What are the advantages of Servlet chains?

Servlet chains have the following advantages:

- Servlet chains can be undone easily. This helps in quickly reversing the change.
- Servlet chains dynamically handle content that is created. Because of this, one can trust that all our restrictions are maintained, that the special tags are replaced, and even in the output of a servlet, all the dynamically converted PostScript images are properly displayed.
- Servlet chains cache the content for later, so it does not execute the script every time got added.

16. Explain the Servlet Life Cycle.

One of the most striking features of servlets is the Servlet Life Cycle. This is a powerful mixture of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The CGI has certain resource and performance problems. In low-level server API programming, there are some security concerns as well. These are addressed by the servlet engines by the servlet life cycle. A servlet engine might execute all of its servlets in a single Java virtual machine (JVM). Servlets can efficiently share data with each other as they share the same JVM. Still, they are prevented from accessing each other's private data by the Java language. Additionally, servlets can be permitted to remain between requests as object instances. Thus they take up lesser memory than the complete processes.

17. What is the life cycle contract that a servlet engine must conform to?

The life cycle contract that a servlet engine must conform to is as follows:

- Create the servlet and initialize it.
- Manage none or more calls for service from clients.
- Destroy the servlet and then the garbage collects it.

18. What do you mean by Servlet Reloading?

Servlet reloading may appear to be a simple feature, but it's actually quite a trick—and requires quite a hack. The objects in ClassLoader are developed to load a class just once. To solve this limitation and to load servlets multiple times, servers use custom class loaders. These custom class loaders load servlets from the default servlets directory.

When a server dispatches a request to a servlet, it first checks if the servlet's class file has changed on disk. If the change appears, then the server abandons the class that the loader used to load the old version and then creates a new instance of the custom class loader to load the new version. Old servlet versions can stay in memory indefinitely, but the old versions are not used to handle any more requests.

19. What are the methods that a servlet can use to get information about the server?

A servlet can be used to learn about its server using 4 different methods. Out of these, two methods are called using the ServletRequest object. These are passed to the servlet. The other two are called from the ServletContext object. In these, the servlet is executing.

20. How can a servlet get the name of the server and the port number for a particular request?

A servlet can get the name of the server and the port number for a particular request with `getServerName()` and `getServerPort()`, respectively:

```
public String ServletRequest.getServerName()
public int ServletRequest.getServerPort()
```

These methods are attributes of ServletRequest because the values can change for different requests if the server has more than one name (a technique called virtual hosting).

The `getServerInfo()` and `getAttribute()` methods of ServletContext supply information about the server software and its attributes:

```
public String ServletContext.getServerInfo()
public Object ServletContext.getAttribute(String name)
```

21. How can a servlet get information about the client machine?

A servlet can use `getRemoteAddr()` and `getRemoteHost()` to retrieve the IP address and hostname of the client machine, respectively:

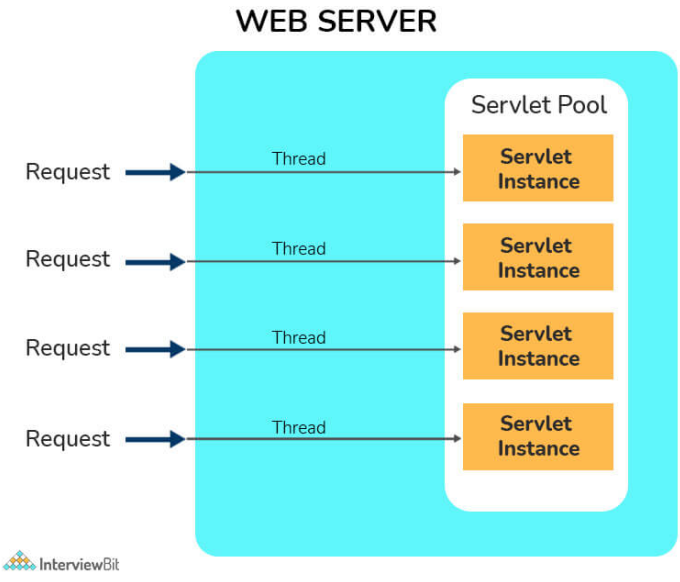
```
public String ServletRequest.getRemoteAddr()
public String ServletRequest.getRemoteHost()
```

Both values are returned as String objects.

22. Explain the Single-Thread Model in servlets.

It is standard to have a single servlet instance for each registered name of the servlet. However, instead of this, it is also possible for a servlet to choose to have a pool of instances created for each of its names that all share the task of handling requests. These servlets indicate this action by implementing the `javax.servlet.SingleThreadModel` interface.

According to the Servlet API documentation, a server loading the SingleThreadModel servlet should guarantee, “that no two threads will execute concurrently the service method of that servlet.” Each thread uses a free servlet instance from the pool in order to achieve this. Therefore, any servlet using the SingleThreadModel isn’t needed to synchronize usage to its instance variables and is considered thread-safe.



23. How does Background Processing take place in servlets?

Servlets can do more than just persist between the accesses. They can also execute between accesses. A thread that has been started by a servlet can continue to execute even after the response has been sent. This ability proves most useful for the tasks that are long-running, and whose incremental results should be made available to multiple clients. A background thread that has been started in init() performs continuous work. It also performs request-handling threads displaying the current status with doGet().

24. How does Servlet collaboration take place?

Servlets running together in the same server have many ways to communicate with one another. There are two main styles of servlet collaboration:

- **Sharing information:** Sharing information involves two or more servlets sharing the state or even resources. A special case of sharing information is Session tracking.
- **Sharing control:** Sharing control involves two or more servlets sharing control of the request. For example, one servlet could receive the request but let another servlet handle some or all of the request-handling responsibilities.

25. Explain Request parameters associated with servlets.

There can be any variety of request parameters related to the servlet with every access to it. These parameters are usually [name-value] pairs that give the servlet any further information that it desires so as to handle the request.

An HTTP servlet gets its request parameters as a part of its query string or as encoded post data. A servlet used as a server-side includes its parameters equipped with PARAM tags.

Fortunately, although a servlet will receive parameters in an exceeding variety of various ways, every servlet retrieves its parameters the same way, by using `getParameter()` and `getParameterValues()` :

```
public String ServletRequest.getParameter(String name)
public String[] ServletRequest.getParameterValues(String name)
```

26. What are the three methods of inter-servlet communication?

The three methods of inter servlet communication are:

- **Servlet manipulation:** In Servlet manipulation, one servlet directly invokes the methods of another. These servlets can get references to other servlets using getServletNames() and getServlet(String name).
- **Servlet reuse:** In Servlet reuse, one servlet uses another's abilities for its own purposes. In some cases, this requires forcing a servlet load using a manual HTTP request.
- **Servlet collaboration:** In Servlet collaboration, the cooperating servlets share information. Servlets can share information using the system properties list, using a shared object, or using inheritance.

27. What are the reasons we use inter-servlet communication?

There are three major reasons to use the inter servlet communication:

- Direct servlet manipulation
- Servlet reuse
- Servlet collaboration

28. What do you mean by Servlet Manipulation?

When one servlet accesses the loaded servlets on its server, it is called Servlet Manipulation. It also optionally performs some task on one or more of them. A servlet gets information about other servlets through the ServletContext object. We use getServlet() to get a particular servlet:

```
public Servlet ServletContext.getServlet(String name) throws ServletException
```

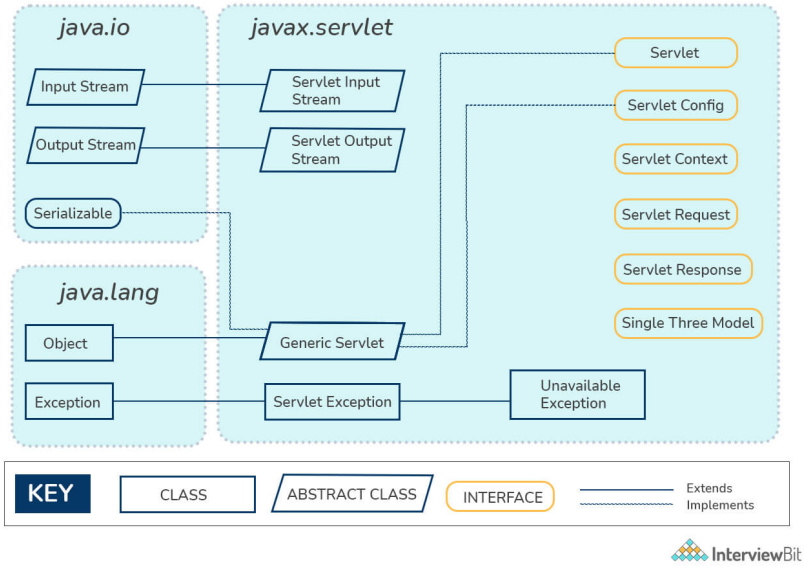
29. What is the javax.servlet package?

The core of the Servlet API is the javax.servlet package. It includes the basic Servlet interface, which all servlets must implement in one form or another, and an abstract GenericServlet class for developing basic servlets.

This package comprises of the following:

- Classes for communicating with the host server and client (ServletRequest and ServletResponse)
- Communicating with the client (ServletInputStream and ServletOutputStream).

In situations where the underlying protocol is unknown, servlets should confine themselves to the classes within this package.



Conclusion:

Unlike CGI and FastCGI, which use many processes to handle separate programs and separate requests, servlets are all handled by separate threads within the webserver process. Thus, the servlets are efficient and scalable. As servlets run within the web server, they can interact very closely with the server to do things that are not possible with CGI scripts.

An advantage of servlets is that they are portable: both across operating systems like with Java and also across web servers.

All of the major web servers support servlets.

References and Resources:

- Java Servlet Programming, by Jason Hunter, Published by O'Reilly Media, Inc.
- Java Servlet & JSP Cookbook, by Bruce W. Perry
- Java Interview
- JSP Interview

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions