

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)[HOME](#) » [INTERVIEW QUESTIONS](#) » 50 SERVLET INTERVIEW QUESTIONS AND ANSWERS

# 50 Servlet Interview Questions and Answers

APRIL 4, 2018 BY [PANKAJ](#) — [85 COMMENTS](#)

**Servlets** are very important topic of Java EE and all of the web applications framework such as Spring and Struts are built on top of it. This makes servlet interview questions a hot topic in interviews.

Here I am providing a list of 50 **servlet interview questions** with answers to help you tackle most of the interview questions related to servlets and web applications in java.



## Servlet Interview Questions

1. What is different between web server and application server?
2. Which HTTP method is non-idempotent?
3. What is the difference between GET and POST method?
4. What is MIME Type?
5. What is a web application and what is its directory structure?
6. What is a servlet?
7. What are the advantages of Servlet over CGI?
8. What are common tasks performed by Servlet Container?
9. What is ServletConfig object?
10. What is ServletContext object?
11. What is difference between ServletConfig and ServletContext?
12. What is Request Dispatcher?
13. What is difference between PrintWriter and ServletOutputStream?

14. Can we get PrintWriter and ServletOutputStream both in a servlet?
15. How can we create deadlock situation in servlet?
16. What is the use of servlet wrapper classes?
17. What is SingleThreadModel interface?
18. Do we need to override service() method?
19. Is it good idea to create servlet constructor?
20. What is difference between GenericServlet and HttpServlet?
21. What is the inter-servlet communication?
22. Are Servlets Thread Safe? How to achieve thread safety in servlets?
23. What is servlet attributes and their scope?
24. How do we call one servlet from another servlet?
25. How can we invoke another servlet in a different application?
26. What is difference between ServletResponse sendRedirect() and RequestDispatcher forward() method?
27. Why HttpServlet class is declared abstract?
28. What are the phases of servlet life cycle?
29. What are life cycle methods of a servlet?
30. why we should override only no-args init() method.
31. What is URL Encoding?
32. What are different methods of session management in servlets?
33. What is URL Rewriting?
34. How does Cookies work in Servlets?
35. How to notify an object in session when session is invalidated or timed-out?
36. What is the difference between encodeRedirectUrl and encodeURL?
37. Why do we have servlet filters?
38. What is the effective way to make sure all the servlets are accessible only when user has a valid session?
39. Why do we have servlet listeners?

40. How to handle exceptions thrown by application with another servlet?
41. What is a deployment descriptor?
42. How to make sure a servlet is loaded at the application startup?
43. How to get the actual path of servlet in server?
44. How to get the server information in a servlet?
45. Write a servlet to upload file on server.
46. How do we go with database connection and log4j integration in servlet?
47. How to get the IP address of client in servlet?
48. What are important features of Servlet 3?
49. What are different ways for servlet authentication?
50. How can we achieve transport layer security for our web application?

## Servlet Interview Questions and Answers

### 1. What is different between web server and application server?

A web server responsibility is to handler HTTP requests from client browsers and respond with HTML response. A web server understands HTTP language and runs on HTTP protocol.

Apache Web Server is kind of a web server and then we have specific containers that can execute servlets and JSPs known as servlet container, for example Tomcat.

Application Servers provide additional features such as Enterprise JavaBeans support, JMS Messaging support, Transaction Management etc. So we can say that Application server is a web server with additional functionalities to help developers with enterprise applications.

### 2. Which HTTP method is non-idempotent?

A HTTP method is said to be idempotent if it returns the same result every time. HTTP methods GET, PUT, DELETE, HEAD, and OPTIONS are idempotent method and we should implement our application

to make sure these methods always return same result. HTTP method POST is non-idempotent method and we should use post method when implementing something that changes with every request.

For example, to access an HTML page or image, we should use GET because it will always return the same object but if we have to save customer information to database, we should use POST method. Idempotent methods are also known as safe methods and we don't care about the repetitive request from the client for safe methods.

### 3. **What is the difference between GET and POST method?**

- GET is a safe method (idempotent) where POST is non-idempotent method.
- We can send limited data with GET method and it's sent in the header request URL whereas we can send large amount of data with POST because it's part of the body.
- GET method is not secure because data is exposed in the URL and we can easily bookmark it and send similar request again, POST is secure because data is sent in request body and we can't bookmark it.
- GET is the default HTTP method whereas we need to specify method as POST to send request with POST method.
- Hyperlinks in a page uses GET method.

### 4. **What is MIME Type?**

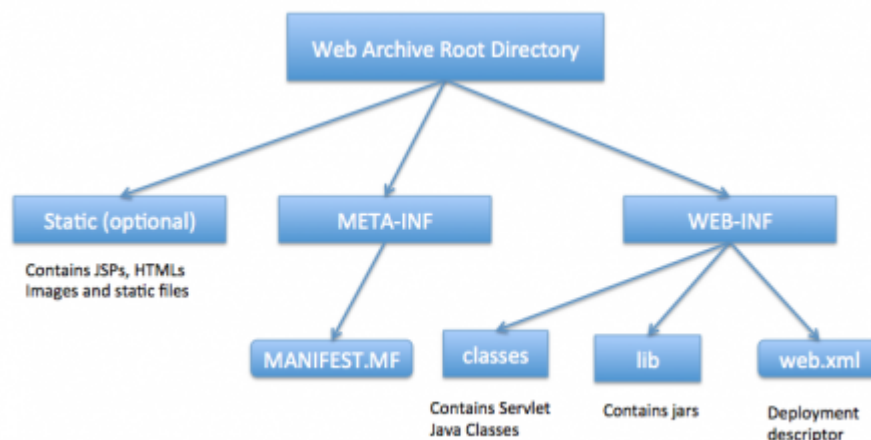
The "Content-Type" response header is known as MIME Type. Server sends MIME type to client to let them know the kind of data it's sending. It helps client in rendering the data for user. Some of the mostly used mime types are text/html, text/xml, application/xml etc.

We can use `ServletContext getMimeType()` method to get the correct MIME type of the file and use it to set the response content type. It's very useful in downloading file through servlet from server.

## 5. What is a web application and what is its directory structure?

Web Applications are modules that run on server to provide both static and dynamic content to the client browser. Apache web server supports PHP and we can create web application using PHP. Java provides web application support through Servlets and JSPs that can run in a servlet container and provide dynamic content to client browser.

**Java Web Applications** are packaged as Web Archive (WAR) and it has a defined structure like below image.



Read more about web applications at [Java Web Application](https://www.journaldev.com/2015/servlet-interview-questions-and-answers).

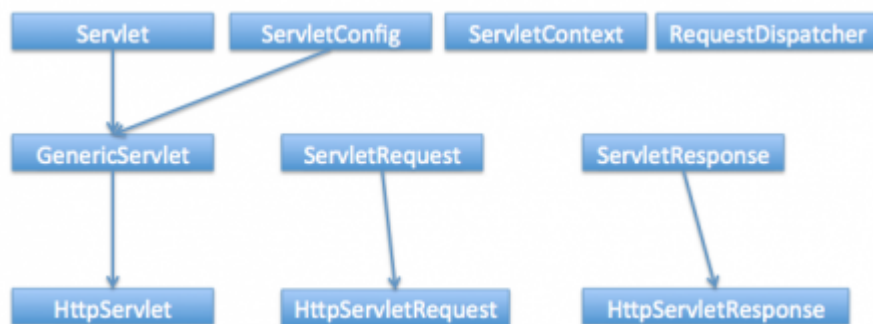
## 6. What is a servlet?

Java Servlet is server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.

The `javax.servlet` and `javax.servlet.http` packages provide interfaces and classes for writing our own servlets.

All servlets must implement the `javax.servlet.Servlet` interface, which defines servlet lifecycle methods. When implementing a generic service, we can extend the `GenericServlet` class provided with the Java Servlet API. The `HttpServlet` class provides methods, such as `doGet()` and `doPost()`, for handling HTTP-specific services.

Most of the times, web applications are accessed using HTTP protocol and thats why we mostly extend `HttpServlet` class. Servlet API hierarchy is shown in below image.



Read more at [Servlet Tutorial](#).

## 7. What are the advantages of Servlet over CGI?

Servlet technology was introduced to overcome the shortcomings of CGI technology.

- Servlets provide better performance than CGI in terms of processing time, memory utilization because servlets use benefits of **multithreading** and for each request a new thread is created, that is faster than loading creating new Object for each request with CGI.
- Servlets are platform and system independent, the web application developed with Servlet can be run on any standard web container such as Tomcat, JBoss, Glassfish servers and on operating systems such as Windows, Linux, Unix, Solaris, Mac etc.
- Servlets are robust because container takes care of life cycle of servlet and we don't need to worry about memory leaks, security, garbage collection etc.
- Servlets are maintainable and learning curve is small because all we need to take care is business logic for our application.

## 8. 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. Because of container, we don't need to build a server socket to listen for any request from web client, parse the request and generate response. All these important and complex tasks are done by container and all we need to focus is on business logic for the applications.
- **Lifecycle and Resource Management:** Servlet Container takes care of managing the life cycle of servlet. 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.

## 9. What is ServletConfig object?

`javax.servlet.ServletConfig` is used to pass configuration information to Servlet. Every servlet has it's own **ServletConfig** object and servlet container is responsible for instantiating this object. We can provide servlet init parameters in web.xml file or through use of `WebInitParam` annotation. We can use `getServletConfig()` method to get the ServletConfig object of the servlet.

## 10. What is ServletContext 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. Servlet containers may also provide context objects that are unique to a group of servlets and which is tied to a specific portion of the URL path namespace of the host.

ServletContext is enhanced in Servlet Specs 3 to introduce methods through which we can programmatically add Listeners and Filters and Servlet to the application. It also provides some utility methods such as `getMimeType()`, `getResourceAsStream()` etc.

## 11. What is difference between ServletConfig and ServletContext?

Some of the differences between ServletConfig and ServletContext are:

- ServletConfig is a unique object per servlet whereas ServletContext is a unique object for complete application.
- ServletConfig is used to provide init parameters to the servlet whereas ServletContext is used to provide application level init parameters that all other servlets can use.
- We can't set attributes in ServletConfig object whereas we can set attributes in ServletContext that other servlets can use in their implementation.

## 12. What is Request Dispatcher?

RequestDispatcher interface is used to forward the request to another resource that can be HTML, JSP or another servlet in same application. We can also use this to include the content of another resource to the response. This interface is used for inter-servlet communication in the same context.

There are two methods defined in this interface:

1. void forward(ServletRequest request, ServletResponse response) – forwards the request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
2. void include(ServletRequest request, ServletResponse response) – includes the content of a resource (servlet, JSP page, HTML file) in the response.

We can get RequestDispatcher in a servlet using ServletContext getRequestDispatcher(String path) method. The path must begin with a / and is interpreted as relative to the current context root.

### 13. **What is difference between PrintWriter and ServletOutputStream?**

PrintWriter is a character-stream class whereas ServletOutputStream is a byte-stream class. We can use PrintWriter to write character based information such as character array and String to the response whereas we can use ServletOutputStream to write byte array data to the response.

We can use ServletResponse `getWriter()` to get the PrintWriter instance whereas we can use ServletResponse `getOutputStream()` method to get the ServletOutputStream object reference.

You can read more about IO in java at [Java IO Tutorial](#).

### 14. **Can we get PrintWriter and ServletOutputStream both in a servlet?**

We can't get instances of both PrintWriter and ServletOutputStream in a single servlet method, if we invoke both the methods; `getWriter()` and `getOutputStream()` on response; we will get `java.lang.IllegalStateException` at runtime with message as other method has already been called for this response.

### 15. **How can we create deadlock situation in servlet?**

We can create deadlock in servlet by making a loop of method invocation, just call `doPost()` method from `doGet()` method and `doGet()` method to `doPost()` method to create deadlock situation in servlet.

Read more about deadlock in multithreading at [Java Deadlock Example](#).

### 16. **What is the use of servlet wrapper classes?**

Servlet HTTP API provides two wrapper classes – `HttpServletRequestWrapper` and `HttpServletResponseWrapper`. These wrapper classes are provided to help developers with custom implementation of servlet request and response types. We can extend these classes and override only specific methods we need to implement for custom request and response objects. These classes are not used in normal servlet programming.

### 17. **What is SingleThreadModel interface?**

SingleThreadModel interface was provided for thread safety and it guarantees that no two threads will execute concurrently in the servlet's service method. However SingleThreadModel does not solve all thread safety issues. For example, session attributes and static variables can still be accessed by multiple requests on multiple threads at the same time, even when SingleThreadModel servlets are used. Also it takes out all the benefits of multithreading support of servlets, that's why this interface is deprecated in Servlet 2.4.

### 18. **Do we need to override service() method?**

When servlet container receives client request, it invokes the `service()` method which in turn invokes the `doGet()`, `doPost()` methods based on the HTTP method of request. I don't see any use case where we would like to override `service()` method. The whole purpose of `service()` method is to forward to request to corresponding HTTP method implementations. If we have to do some pre-processing of request, we can always use servlet filters and listeners.

### 19. **Is it good idea to create servlet constructor?**

We can define a constructor for servlet but I don't think it's of any use because we won't be having access to the `ServletConfig` object until unless servlet is initialized by container. Ideally if we have to initialize any resource for servlet, we should override `init()` method where we can access servlet init parameters using `ServletConfig` object.

## 20. **What is difference between GenericServlet and HttpServlet?**

GenericServlet is protocol independent implementation of Servlet interface whereas HttpServlet is HTTP protocol specific implementation. Most of the times we use servlet for creating web application and that's why we extend HttpServlet class. HttpServlet class extends GenericServlet and also provide some other methods specific to HTTP protocol.

## 21. **What is the inter-servlet communication?**

When we want to invoke another servlet from a servlet service methods, we use inter-servlet communication mechanisms. We can invoke another servlet using `RequestDispatcher forward()` and `include()` methods and provide additional attributes in request for other servlet use.

## 22. **Are Servlets Thread Safe? How to achieve thread safety in servlets?**

HttpServlet `init()` method and `destroy()` method are called only once in servlet life cycle, so we don't need to worry about their synchronization. But service methods such as `doGet()` or `doPost()` are getting called in every client request and since servlet uses multithreading, we should provide thread safety in these methods.

If there are any local variables in service methods, we don't need to worry about their thread safety because they are specific to each thread but if we have a shared resource then we can use synchronization to achieve thread safety in servlets when working with shared resources.

The thread safety mechanisms are similar to thread safety in standalone java application, read more about them at [Thread Safety in Java](#).

### 23. **What is servlet attributes and their scope?**

Servlet attributes are used for inter-servlet communication, we can set, get and remove attributes in web application. There are three scopes for servlet attributes – request scope, session scope and application scope.

ServletRequest, HttpSession and ServletContext interfaces provide methods to get/set/remove attributes from request, session and application scope respectively.

Servlet attributes are different from init parameters defined in web.xml for ServletConfig or ServletContext.

### 24. **How do we call one servlet from another servlet?**

We can use RequestDispatcher forward() method to forward the processing of a request to another servlet. If we want to include the another servlet output to the response, we can use RequestDispatcher include() method.

### 25. **How can we invoke another servlet in a different application?**

We can't use RequestDispatcher to invoke servlet from another application because it's specific for the application. If we have to forward the request to a resource in another application, we can use ServletResponse sendRedirect() method and provide complete URL of another servlet. This sends the response to client with response code as 302 to forward the request to another URL. If we have to send some data also, we can use cookies that will be part of the servlet response and sent in the request to another servlet.

## 26. What is difference between `ServletResponse sendRedirect()` and `RequestDispatcher forward()` method?

1. `RequestDispatcher forward()` is used to forward the same request to another resource whereas `ServletResponse sendRedirect()` is a two step process. In `sendRedirect()`, web application returns the response to client with status code 302 (redirect) with URL to send the request. The request sent is a completely new request.
2. `forward()` is handled internally by the container whereas `sendRedirect()` is handled by browser.
3. We should use `forward()` when accessing resources in the same application because it's faster than `sendRedirect()` method that required an extra network call.
4. In `forward()` browser is unaware of the actual processing resource and the URL in address bar remains same whereas in `sendRedirect()` URL in address bar change to the forwarded resource.
5. `forward()` can't be used to invoke a servlet in another context, we can only use `sendRedirect()` in this case.

## 27. Why `HttpServlet` class is declared abstract?

`HttpServlet` class provide HTTP protocol implementation of servlet but it's left abstract because there is no implementation logic in service methods such as `doGet()` and `doPost()` and we should override at least one of the service methods. That's why there is no point in having an instance of `HttpServlet` and is declared abstract class.

Read more about [abstract class](#).

## 28. What are the phases of servlet life cycle?

We know that Servlet Container manages the life cycle of Servlet, there are four phases of servlet life cycle.

1. Servlet Class Loading – When container receives request for a servlet, it first loads the class into memory and calls its default no-args constructor.
2. Servlet Class Initialization – Once the servlet class is loaded, container initializes the ServletContext object for the servlet and then invoke its init method by passing servlet config object. This is the place where a servlet class transforms from normal class to servlet.
3. Request Handling – Once servlet is initialized, its ready to handle the client requests. For every client request, servlet container spawns a new thread and invokes the service() method by passing the request and response object reference.
4. Removal from Service – When container stops or we stop the application, servlet container destroys the servlet class by invoking its destroy() method.

## 29. **What are life cycle methods of a servlet?**

Servlet Life Cycle consists of three methods:

1. public void init(ServletConfig config) – This method is used by container to initialize the servlet, this method is invoked only once in the lifecycle of servlet.
2. public void service(ServletRequest request, ServletResponse response) – This method is called once for every request, container can't invoke service() method until unless init() method is executed.
3. public void destroy() – This method is invoked once when servlet is unloaded from memory.

## 30. **why we should override only no-args init() method.**

If we have to initialize some resource before we want our servlet to process client requests, we should override init() method. If we override init(ServletConfig config) method, then the first statement should be super(config) to make sure superclass init(ServletConfig config) method is invoked first. That's why GenericServlet provides another helper init() method without argument that gets called at the end of



init(ServletConfig config) method. We should always utilize this method for overriding init() method to avoid any issues as we may forget to add super() call in overriding init method with ServletConfig argument.

### 31. **What is URL Encoding?**

URL Encoding is the process of converting data into CGI form so that it can travel across the network without any issues. URL Encoding strip the white spaces and replace special characters with escape characters. We can use `java.net.URLEncoder.encode(String str, String unicode)` to encode a String. URL Decoding is the reverse process of encoding and we can use `java.net.URLDecoder.decode(String str, String unicode)` to decode the encoded string. For example "Pankaj's Data" is encoded to "Pankaj%27s+Data".

### 32. **What are different methods of session management in servlets?**

Session is a conversational state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

Some of the common ways of session management in servlets are:

1. User Authentication
2. HTML Hidden Field
3. Cookies
4. URL Rewriting
5. Session Management API

Read more about these session management approaches in detail at **Servlet Session Management Tutorial**.

### 33. What is URL Rewriting?

We can use HttpSession for session management in servlets but it works with Cookies and we can disable the cookie in client browser. Servlet API provides support for URL rewriting that we can use to manage session in this case.

The best part is that from coding point of view, it's very easy to use and involves one step – encoding the URL. Another good thing with Servlet URL Encoding is that it's a fallback approach and it kicks in only if browser cookies are disabled.

We can encode URL with HttpServletResponse encodeURL() method and if we have to redirect the request to another resource and we want to provide session information, we can use encodeRedirectURL() method.

Read More at [Servlet URL Rewriting](#).

### 34. How does Cookies work in Servlets?

Cookies are used a lot in web client-server communication, it's not something specific to java. Cookies are text data sent by server to the client and it gets saved at the client local machine.

Servlet API provides cookies support through javax.servlet.http.Cookie class that implements Serializable and Cloneable interfaces.

HttpServletRequest getCookies() method is provided to get the array of Cookies from request, since there is no point of adding Cookie to request, there are no methods to set or add cookie to request.

Similarly HttpServletResponse addCookie(Cookie c) method is provided to attach cookie in response header, there are no getter methods for cookie.

Read more at [Cookies in Servlets](#).

### 35. **How to notify an object in session when session is invalidated or timed-out?**

If we have to make sure an object gets notified when session is destroyed, the object should implement `javax.servlet.http.HttpSessionBindingListener` interface. This interface defines two callback methods – `valueBound()` and `valueUnbound()` that we can define to implement processing logic when the object is added as attribute to the session and when session is destroyed.

Recommended reading [Servlet Listener](#).

### 36. **What is the difference between encodeRedirectUrl and encodeURL?**

HttpServletResponse provide method to encode URL in HTML hyperlinks so that the special characters and white spaces are escaped and append session id to the URL. It behaves similar to URLEncoder encode method with additional process to append jsessionid parameter at the end of the URL.

However HttpServletResponse encodeRedirectUrl() method is used specially for encode the redirect URL in response.

So when we are providing URL rewriting support, for hyperlinks in HTML response, we should use `encodeURL()` method whereas for redirect URL we should use `encodeRedirectUrl()` method.

### 37. **Why do we have servlet filters?**

Servlet Filters are pluggable java components that we can use to intercept and process requests before they are sent to servlets and response after servlet code is finished and before container sends the response back to the client.

Some common tasks that we can do with filters are:

- Logging request parameters to log files.
- Authentication and authentication of request for resources.
- Formatting of request body or header before sending it to servlet.
- Compressing the response data sent to the client.
- Alter response by adding some cookies, header information etc.

Read more about filters at [Servlet Filter](#).

### 38. **What is the effective way to make sure all the servlets are accessible only when user has a valid session?**

We know that servlet filters can be used to intercept request between servlet container and servlet, we can utilize it to create authentication filter and check if request contains a valid session or not.

Check out Authentication Filter example at [Servlet Filter Example](#).

### 39. **Why do we have servlet listeners?**

We know that using ServletContext, we can create an attribute with application scope that all other servlets can access but we can initialize ServletContext init parameters as String only in deployment descriptor (web.xml). What if our application is database oriented and we want to set an attribute in ServletContext for Database Connection.

If you application has a single entry point (user login), then you can do it in the first servlet request but if we have multiple entry points then doing it everywhere will result in a lot of code redundancy. Also if database is down or not configured properly, we won't know until first client request comes to server. To handle these scenario, servlet API provides Listener interfaces that we can implement and configure to listen to an event and do certain operations.

Read more about different types of listeners and example at [Servlet Listener](#).

#### 40. **How to handle exceptions thrown by application with another servlet?**

If you notice, doGet() and doPost() methods throw ServletException and IOException. Since browser understand only HTML, when our application throw exception, servlet container processes the exception and generate a HTML response. Same goes with other error codes like 404, 403 etc.

Servlet API provides support for custom Exception and Error Handler servlets that we can configure in deployment descriptor, the whole purpose of these servlets are to handle the Exception or Error raised by application and send HTML response that is useful for the user. We can provide link to application home page or some details to let user know what went wrong.

We can configure them in web.xml like below:

```
<error-page>
    <error-code>404</error-code>
    <location>/AppExceptionHandler</location>
</error-page>

<error-page>
    <exception-type>javax.servlet.ServletException</exception-type>
    <location>/AppExceptionHandler</location>
</error-page>
```

Read more at [Servlet Exception Handling](#).

#### 41. What is a deployment descriptor?

Deployment descriptor is a configuration file for the web application and its name is web.xml and it resides in WEB-INF directory. Servlet container uses this file to configure web application servlets, servlet config params, context init params, filters, listeners, welcome pages and error handlers.

With servlet 3.0 annotations, we can remove a lot of clutter from web.xml by configuring servlets, filters and listeners using annotations.

#### 42. How to make sure a servlet is loaded at the application startup?

Usually servlet container loads a servlet on the first client request but sometimes when the servlet is heavy and takes time to load, we might want to load it on application startup. We can use load-on-startup element with servlet configuration in web.xml file or use@WebServlet annotation loadOnStartup variable to tell container to load the servlet on system startup.

```
<servlet>
    <servlet-name>foo</servlet-name>
    <servlet-class>com.foo.servlets.Foo</servlet-class>
    <load-on-startup>5</load-on-startup>
</servlet>
```

The load-on-startup value should be int, if it's negative integer then servlet container will load the servlet based on client requests and requirement but if it's 0 or positive, then container will load it on application startup.

If there are multiple servlets with load-on-startup value such as 0,1,2,3 then lower integer value servlet will be loaded first.

#### 43. **How to get the actual path of servlet in server?**

We can use following code snippet to get the actual path of the servlet in file system.

```
getServletContext().getRealPath(request.getServletPath())
```

#### 44. **How to get the server information in a servlet?**

We can use below code snippet to get the servlet information in a servlet through servlet context object.

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

#### 45. **Write a servlet to upload file on server.**

File Upload and Download and common tasks in a java web application. Unfortunately Servlet API doesn't provide easy methods to upload file on server, so we can use Apache FileUpload jar to make our life easier.

Please read [File Upload Servlet](#) post that provide all the necessary details with example program to upload and download file using servlets.

#### 46. **How do we go with database connection and log4j integration in servlet?**

If you work with database connection a lot in your web application, its best to initialize it in a servlet context listener and set it as a context attribute for other servlets to use.

Integrating Log4j is also very easy in web applications, all we need is a log4j configuration XML or property file and then configure it in a servlet context listener.

For complete example, please look into [Servlet Database and Log4j Example](#).

#### 47. **How to get the IP address of client in servlet?**

We can use `request.getRemoteAddr()` to get the client IP address in servlet.

#### 48. **What are important features of Servlet 3?**

Servlet Specs 3.0 was a major release and some of the important features are:

1. **Servlet Annotations:** Prior to Servlet 3, all the servlet mapping and it's init parameters were used to defined in web.xml, this was not convenient and more error prone when number of servlets are huge in an application.



Servlet 3 introduced use of java annotations to define a servlet, filter and listener servlets and init parameters. Some of the important Servlet API annotations are `WebServlet`, `WebInitParam`, `WebFilter` and `WebListener`. Read more about them at [Servlet 3 annotations](#).

2. **Web Fragments:** Prior to servlet specs 3.0, all the web application configurations are required to be present in the `web.xml` that makes it cluttered with lot of elements and chances of error increases. So servlet 3 specs introduced web fragments where we can have multiple modules in a single web application, all these modules should have `web-fragment.xml` file in `META-INF` directory. We can include all the elements of `web.xml` inside the `web-fragment.xml` too. This helps us in dividing our web application into separate modules that are included as JAR file in the web application lib directory.
3. **Adding Web Components dynamically:** We can use `ServletContext` object to add servlets, filters and listeners programmatically. This helps us in building dynamic system where we are loading a component only if we need it. These methods are `addServlet()`, `addFilter()` and `addListener()` defined in the servlet context object.
4. **Asynchronous Processing:** Asynchronous support was added to delegate the request processing to another thread rather than keeping the servlet thread busy. It can increase the throughput performance of the application. This is an advance topic and I recommend to read [Async Servlet](#) tutorial.

## 49. What are different ways for servlet authentication?

Servlet Container provides different ways of login based servlet authentication:

1. **HTTP Basic Authentication**
2. **HTTP Digest Authentication**
3. **HTTPS Authentication**

4. **Form Based Login:** A standard HTML form for authentication, advantage is that we can change the login page layout as our application requirements rather than using HTTP built-in login mechanisms.

## 50. **How can we achieve transport layer security for our web application?**

We can configure our servlet container to use SSL for message communication over the network. To configure SSL on Tomcat, we need a digital certificate that can be created using Java keytool for development environment. For production environment, you should get the digital certificate from SSL certificate providers, for example, Verisign or Entrust.

Read more at [Configure SSL on Tomcat](#) article.

That's all for the servlet interview questions and answers. I will keep on adding more servlet based interview questions to the list in future, so don't forget to bookmark it for future reference.

Please share your thoughts in comments and share your love with sharing on Google Plus, Facebook or Twitter. □

**Update:** If you liked these questions, I am sure you will like [JSP Interview Questions](#) too.

References:

- <http://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html>
- [https://en.wikipedia.org/wiki/Java\\_servlet](https://en.wikipedia.org/wiki/Java_servlet)

---

FILED UNDER: [INTERVIEW QUESTIONS, JAVA EE](#)

### About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

[« Async Servlet Example](#)[JSP Example Tutorial for Beginners »](#)

## Comments

**pushpendra patel says**

JUNE 20, 2018 AT 9:06 AM

Add some more questions like SQL injection and double posting problem

[Reply](#)**Ram says**

MARCH 15, 2018 AT 9:17 AM

Hi Pankaj, I observed a typo mistake possibly.

In the 1st question's answer,

"A web server responsibility is to handler HTTP requests from client browsers and respond with HTML response. A web server understands HTTP language and runs on HTTP protocol".

The last statement states that "A web server understands HTTP language", rather It should be "A web server understands HTML language ". HTTP is not a language as you have mentioned. Correct me If I'm wrong.

[Reply](#)**harsha says**

NOVEMBER 5, 2017 AT 2:54 PM

Hi

can you add the questions on enterprise java bean (EJB ) tutorial and interview questions

[Reply](#)

**Sureshg says**

JULY 15, 2017 AT 5:17 AM

Good one to read !!

[Reply](#)

**Aman says**

JULY 12, 2017 AT 7:45 AM

Great post

[Reply](#)

**Diganta Banerjee says**

JUNE 26, 2017 AT 11:42 PM

Hi Pankaj,

In Q#32 the URL to Servlet Session Management Tutorial. is broken. Please fix this.

Regards,

Diganta

[Reply](#)

**Arvind kumar Yadav says**

JANUARY 4, 2017 AT 4:01 AM

excellent tutorials .. thanks

[Reply](#)**Aditya Verma says**

NOVEMBER 17, 2016 AT 10:56 AM

Hi Pankaj,

Thanks for such an awesome Post.

In the question "What are the phases of servlet life cycle?" between loading and initialization there is one more Instantiation phase also.

Please refer the link: <http://docs.oracle.com/javaee/6/tutorial/doc/bnaf1.html>

Regards,

Aditya Verma

[Reply](#)**Aditya Verma says**

NOVEMBER 10, 2016 AT 10:49 AM

In the answer to question no 19 you wrote that we can have constructor of a Servlet. But as Servlet is an Interface and we can't have constructor declared for an Interface.

So I think answer is incorrect. Please do correct me if I am wrong

Reply

**Pankaj says**

NOVEMBER 10, 2016 AT 9:22 PM

Mostly when we say Servlet, we mean HttpServlet. Answer is also related to HttpServlet only.

Reply

**Jagadeesh says**

APRIL 17, 2017 AT 11:20 AM

we can define only zero arg constructor only for any user defined servlets. because Server uses Class.forName() and newInstance() methods to create the Servlet Object. Servlet creation will fail if there is no zero arg constructor available in server. that is the reason java people recommended to go with init method instead of Constructor

Reply

**Mohammed Naveed says**

JULY 23, 2016 AT 10:51 PM

It is a great post for the freshers and as well as for the experienced.

Sir can you please provide me with a pdf file of this.

Reply

**Mamatha says**

JULY 22, 2016 AT 11:15 AM

Thanks for spending time for providing this kind of Interview Questions which is helpful for the interview....covered all the topics of Servlets... Thanq.....

[Reply](#)

**shital says**

JULY 18, 2016 AT 11:11 AM

really very helpful....thanks

[Reply](#)

**Pavan says**

MAY 29, 2016 AT 4:53 AM

Very Useful stuff... Thanks..

[Reply](#)

**Pankaj says**

MAY 29, 2016 AT 6:18 AM

Thanks Pavan for kind words.

[Reply](#)



**franci says**

JULY 6, 2016 AT 8:23 PM

really super very usefull.....give some more points on servlet elements

[Reply](#)**Amr says**

MARCH 16, 2016 AT 4:07 AM

very helpful, thanks a lot

[Reply](#)**ravi says**

MARCH 2, 2016 AT 3:30 AM

excellent work by pankaj.....these Q/A provides touches all parts of servlets...really nice ☐

[Reply](#)**Shubham Jaiswal says**

DECEMBER 16, 2015 AT 4:42 PM

Oh !!! Wow

These question are very Helpful because it has covered all the topic and one can clear his/her concept

in JSP and servlet.

The entire material are well organised so that we can have the better understanding about the topics covered...

Thank you very much Author...

[Reply](#)

**sonalichauhan says**

OCTOBER 27, 2015 AT 12:12 PM

Vry imp. Ques for.interview.thnks a lot..

[Reply](#)

**ERIKALA REDDY says**

OCTOBER 15, 2015 AT 2:38 AM

Good servlet questions & answers.Thank you

[Reply](#)

**Rama says**

OCTOBER 11, 2015 AT 10:04 AM

veri nice questions.

These quesions are very helpfull to clear the interview.

Thank u so much.

[Reply](#)

**Digvijay Singh says**

JULY 29, 2015 AT 6:47 AM

Hi Pankaj,

Very nice questions!!

I have a questions which I am struggling with. I would like to share it here.

Servlet container spawns a new thread on every request but as we all know that our servlet doesn't either extends Thread class or implements Runnable interface then how does it works out?

Plz help me here.

Thanks

[Reply](#)

**srinivas rompala says**

SEPTEMBER 13, 2015 AT 7:45 AM

I think this may clear your doubt....

Per request means when an HTTP request is made, a thread is created or retrieved from a pool to serve it. One thread serves the whole request. Thread per connection would be the same thing except the thread is used for an entire connection, which could be multiple requests and could also have a lot of dead time in between requests. Servlet containers are thread per request. There may be some implementations that offer thread per connection, but I don't know, and it seems like it would be very wasteful.

Creating a thread inside another thread doesn't establish any special relationship, and the whole point of doing so in most cases is to let the one thread do more work or terminate while the other thread continues working. In your scenario, using a different thread to do work required by a request

will, as you expect, allow the response to be sent immediately. The thread used to serve that request will also be immediately available for another request, regardless of how long your other thread takes to complete. This is pretty much the way of doing asynchronous work in a thread-per-request servlet container.

[Reply](#)

**vikas kumar singh says**

JULY 17, 2015 AT 10:38 AM

very helpful questions

[Reply](#)

**Gourish Paragi says**

JUNE 27, 2015 AT 11:08 PM

Hi Pankaj,

I love to read the concepts which you have been posted.. please share more videos on Collections and Some important core java Concepts...really i am enjoying your tutorial lot.. easy to read and understand..

Thanks&regards

Gourish P

[Reply](#)

**Sumit Kumar says**

MAY 23, 2015 AT 11:39 PM

Useful questions from interview perspective ...!!!

[Reply](#)

**Rieethika says**

APRIL 22, 2015 AT 10:14 AM

This is really very useful. Thanks a lot !!!

[Reply](#)

**Joseph says**

APRIL 17, 2015 AT 6:27 PM

it touches each and every key point of Servlet Technologies, Very useful. Thank You Very Much.

[Reply](#)

**Lukman says**

MARCH 25, 2015 AT 8:10 AM

very good and helpful...

Thanks..

[Reply](#)

**venkata says**

FEBRUARY 16, 2015 AT 1:03 AM

Hi

it's really helpful ..i have one Questions that how can i configure the multiple servlet java class which is present in .jar .with out configure it is working in JDK1.4 , but not working in JDK1.7.i have around 25 to 30 jar files each jar file i have 30 java class.

where to configure like tomcat/conf/web.xml (or) tomcat/webapps/projecname/web-inf/web.xml.

THQ..

[Reply](#)**dakshina gandikota says**

JANUARY 19, 2015 AT 4:44 PM

Suppose we have two servlets ServletA and Servlet B. ServletA gets a request that requires it to access PrintWriter and then forward to ServletB. Can ServletB access its PrintWriter in this case?

[Reply](#)**Benny says**

JANUARY 16, 2015 AT 9:37 AM

Nice stuff man, really help ful.thaaankss

[Reply](#)

**sahana says**

JANUARY 3, 2015 AT 1:15 AM

The information is very crystal clear and easy to understand.Thanks for the same

[Reply](#)

**Narendra says**

DECEMBER 24, 2014 AT 6:04 AM

Hi Pankaj,

My question is that....Is servlet singleton by default..?

[Reply](#)

**Ramesh says**

JANUARY 15, 2015 AT 8:38 PM

Narendra. Yes, Servlet is singleton by default.

[Reply](#)

**raam says**

FEBRUARY 12, 2015 AT 5:25 AM

yes it is singleton

[Reply](#)

**Dadapeer says**

MARCH 24, 2015 AT 7:11 AM

Singleton – There should always be only one instance and there should be no way of creating multiple instances of the class by any means. If you define a servlet MyServlet extends HttpServlet, you can still go ahead and create any number of instances of this servlet  
`MyServlet ms1 = new MyServlet();`  
`MyServlet ms2 = new MyServlet();`

Servlets are not singleton, for the servlet definition in the web.xml as mentioned below, container creates only one instance of this servlet.

`myservlet1``com.cisco.sdp.ui.common.servlet.LoginServlet`

if you provide again the same configuration for a servlet the container will create another instance of the same servlet. PS- servlet name is myservlet2

`myservlet2``com.cisco.sdp.ui.common.servlet.LoginServlet`

Hence Servlets are not singleton.

Reply

**rama says**

OCTOBER 11, 2015 AT 10:02 AM

Thank u so much.very nice questions. these questions are very helpful to clear the interview.

Reply



**Hanumant Maindad says**

DECEMBER 17, 2014 AT 1:43 PM

all question are very good and i request to provide more question answer.

[Reply](#)

**Laxmin says**

DECEMBER 13, 2014 AT 3:25 AM

Excellent questions ,really helpfull

[Reply](#)

**Sanjana says**

NOVEMBER 17, 2014 AT 1:53 PM

Can some one guide how we can look up for the rest of 30 answers?

Interesting questions.Good for the interviews.

Thanks,

Sanjana.

[Reply](#)

**Jaffar Shariff says**

NOVEMBER 5, 2014 AT 3:24 AM

Really helpful .Thanks

[Reply](#)

### **Rupesh Vislawath says**

OCTOBER 5, 2014 AT 8:46 AM

which describe ways that dynamic information can be made available to all servlet requests sent to an application...?

[Reply](#)

### **Naresh says**

SEPTEMBER 24, 2014 AT 8:41 AM

Q.26 – forward() can't be used to invoke a servlet in another context, we can only use sendRedirect() in this case.

But we can do a

```
ServletContext ctx = request.getServletContext().getContext("/otherapp");
```

```
request.setAttribute("MESSAGE", "Hello There!");
```

```
RequestDispatcher dispatcher = ctx.getRequestDispatcher("/hello.jsp");
```

```
dispatcher.forward(request, response);
```

So can you help understand if the difference you mentioned is correct?

[Reply](#)

**guest says**

SEPTEMBER 18, 2014 AT 10:02 AM

Thank you for posting

[Reply](#)**MOHIT SINGH says**

AUGUST 15, 2014 AT 12:13 AM

Nice stuff ☐

[Reply](#)**puja says**

AUGUST 10, 2014 AT 10:55 AM

Questions and Answers are quite good, appreciable , but it is only 20 in count i am expecting 50.

Thanks

[Reply](#)**Puneet Kaushik says**

AUGUST 7, 2014 AT 3:12 AM

Sir,

Thanks for precious guidance .

[Reply](#)

**Alshad says**

AUGUST 2, 2014 AT 7:13 PM

I am able to see only 20 questions. I was using google chrome in mobile. Thanks for this good info

[Reply](#)

**Anjali Nair says**

JULY 31, 2014 AT 11:13 AM

Thanks a lot for this as well as other interview questions. Very detailed collection, and the tutorial on each topic just made my life easier ☐

[Reply](#)

**Laurenity says**

JULY 26, 2014 AT 1:11 AM

easy to apply sir wooah great

[Reply](#)

**Neha says**

JULY 24, 2014 AT 10:56 AM

Question collection is very nice and to the point. However, I would like to point out in first question's answer, Tomcat is an application server and Apache HTTPD is a web server.

[Reply](#)

**amruta says**

JULY 7, 2014 AT 3:50 AM

Thanx..

[Reply](#)

**kavya jindal says**

JUNE 23, 2014 AT 5:00 PM

Awesome collection.. Thanks a ton!!!

[Reply](#)

**Mohasin Ali says**

MAY 28, 2014 AT 9:40 AM

The questions from 20 are went to hidden, I remembered last time I read all the questions and answers but today when I came here some question went to Hidden (display:none property of style). please remove that hidden attribute.I found this using my Firebug plugin.

By the way nice article

Thanks

Mohasin Ali

[Reply](#)

**The Anonymous says**

JUNE 25, 2014 AT 12:22 AM

is this firebug plugin your's??

[Reply](#)

**Venu says**

MAY 25, 2014 AT 10:52 PM

Really helpful, Covered almost all topics, easy to revise the subject and face the interviews.

[Reply](#)

**Deepak says**

MAY 21, 2014 AT 7:02 PM

Hi Pankaj,

I am only able to see only 20 Servlet Interview Questions and Answers out of 50. Can You please let me know how I can be able to see all the answers .

[Reply](#)

**Tej says**

MAY 3, 2014 AT 8:18 PM

I am following you on twitter, but still the answers are locked for me. How to unlock them?

[Reply](#)

**Pankaj says**

MAY 3, 2014 AT 8:44 PM

Thanks for following, you can close it.

[Reply](#)

**Nagarajan says**

APRIL 28, 2014 AT 7:02 AM

Thank you for your great work.

I really excited to read this page. Nice formatting, clear cut answers.

Wonderful.

[Reply](#)

**Lavesh Singhal says**

APRIL 24, 2014 AT 12:28 AM

Definition of Q 31 doesn't seems a general and correct.

[Reply](#)

**Yashwant Chandrakar says**

APRIL 18, 2014 AT 5:28 AM

awesome sir I really learn a lot from this

[Reply](#)

**Mohamed Zayadi says**

APRIL 13, 2014 AT 4:30 AM

The best article i read ever about Servlet , many thanks Pankaj for the useful information.

Mohammed Zayadi

Huawei Technologies Co.,Ltd

[Reply](#)

**Tarun says**

APRIL 18, 2014 AT 2:02 PM

Lots of thanks and blessing from all job seekers...I

[Reply](#)



**Ravindra Patil says**

MARCH 29, 2014 AT 10:57 AM

plz tell me it is possible to write same function to button in javascript and also in servlet which one is executed and how

Reply

**Pankaj says**

MARCH 29, 2014 AT 6:12 PM

I dont understood what you meant here.

Reply

**Sonia says**

MARCH 22, 2014 AT 8:24 PM

Very informative. Best for interview preparation with basics explained very well. Thanks!! Please do continue.. looking forward to make your website as my goto page for any question in java, web application and much more..

Reply

**braj says**

FEBRUARY 8, 2014 AT 3:22 PM

Kindly review answer given for Question No 42.

Refer link <http://stackoverflow.com/questions/809775/what-does-the-servlet-load-on-startup-value-of-0-zero-signify>

Reply

**Pankaj says**

FEBRUARY 10, 2014 AT 9:49 AM

Thanks Braj for pointing out the mistake, I have verified it again and if the load-on-startup is 0 or positive int, then it gets loaded at the time of server startup. For negative integer, it gets loaded when first client request comes.

Corrected the post.

Reply

**Swarna says**

FEBRUARY 7, 2014 AT 7:16 PM

Amazing.Simple,easy to understand,covers most of the things.Excellent guide for interview preparation!!

Reply

**parvathy says**

FEBRUARY 4, 2014 AT 5:18 PM

very helpful- good ready reckoner – helped to recollect aft span of 2 years

[Reply](#)

**Alpesh Rathod says**

JANUARY 24, 2014 AT 9:59 AM

Awesome post. Covers all the things that are required for interview..Good going ☐

[Reply](#)

**Jag says**

JANUARY 16, 2014 AT 5:29 PM

Please write more...

[Reply](#)

**Rajasekhar D says**

JANUARY 9, 2014 AT 5:12 AM

Hi ,

Can you please help on the below question answers ,

- 1) what is good logic(java or query) to find the login user is valid or not, if database has million records?
- 2) What are requirements to create a customized HashMap ?

Thanks in advance....

[Reply](#)

**HIMANSU NAYAK says**

FEBRUARY 3, 2014 AT 3:05 PM

Most of the Enterprise Application access user details from LDAP, for its structure retrieving user data from LDAP is much faster then a relational database.

[Reply](#)**Raju Mandala says**

JANUARY 3, 2014 AT 4:37 AM

Thank You

[Reply](#)**AlokC says**

DECEMBER 23, 2013 AT 12:38 AM

It was really helpful for my interviews thanks a lot sir...

[Reply](#)**Rakesh says**

DECEMBER 11, 2013 AT 8:49 AM

awesome.... very good and very helpful... Thanks..

[Reply](#)

**chandu says**

DECEMBER 11, 2013 AT 8:17 AM

excellent,entire servlet api is covered. please keep on.  
and request u to revert back to mail if any new post  
thanks a lot

[Reply](#)

**ramanagouda says**

OCTOBER 29, 2013 AT 3:04 PM

really this is useful .....thank you sir.....

[Reply](#)

**Amit says**

AUGUST 24, 2013 AT 12:05 PM

Awesome post, great content and easy to understand... keep it coming.

[Reply](#)

**prakash says**

MAY 15, 2015 AT 4:38 AM

thanks..

[Reply](#)

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT



---

DOWNLOAD ANDROID APP



---

IMPORTANT INTERVIEW QUESTIONS

---

**Java Interview Questions**

- > [Core Java Interview Questions](#)
- > [String Interview Questions](#)
- > [Multithreading Interview Questions](#)
- > [Collections Interview Questions](#)
- > [Exception Interview Questions](#)
- > [Java Programming Interview Questions](#)
- > [Java 8 Interview Questions Part 1](#)
- > [Java 8 Interview Questions Part 2](#)
- > [Servlet Interview Questions](#)
- > [JSP Interview Questions](#)
- > [Struts 2 Interview Questions](#)
- > [JDBC Interview Questions](#)
- > [Spring Interview Questions](#)
- > [Hibernate Interview Questions](#)
- > [JSF Interview Questions](#)
- > [Web Services Interview Questions](#)
- > [Scala Basic Interview Questions](#)
- > [Scala Intermediate Interview Questions](#)
- > [Scala Advanced Interview Questions](#)
- > [Scala Interview Questions Summary](#)
- > [Common Job Interview Questions](#)

## Miscellaneous

- > [Java ClassLoader](#)
- > [String StringBuffer StringBuilder](#)
- > [Java is Pass By Value](#)
- > [Java Heap vs Stack Memory](#)

---

RECOMMENDED TUTORIALS

---



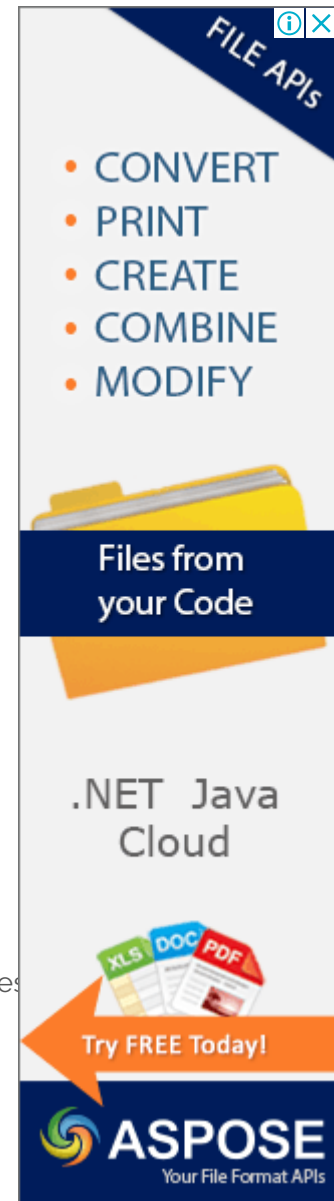
## Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

## Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)
- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

© 2018 · Privacy Policy · Don't copy, it's Bad Karma · Powered by WordPress



FILE APIs

- CONVERT
- PRINT
- CREATE
- COMBINE
- MODIFY

Files from  
your Code

.NET Java  
Cloud

Try FREE Today!

**ASPOSE**  
Your File Format APIs

The advertisement is a vertical banner. At the top, it says 'FILE APIs' in a blue banner. Below this, there is a list of five bullet points: 'CONVERT', 'PRINT', 'CREATE', 'COMBINE', and 'MODIFY'. Under the list is an illustration of a yellow folder. Below the folder is a dark blue banner with the text 'Files from your Code'. This is followed by a light gray section with the text '.NET Java Cloud'. Below that is an illustration of three document icons labeled 'XLS', 'DOC', and 'PDF'. Under these icons is an orange banner with the text 'Try FREE Today!'. At the bottom is a dark blue banner with the ASPOSE logo and the text 'ASPOSE Your File Format APIs'.