1) How many objects of a servlet is created?

Only one object at the time of first request by servlet or web container.

2) Who is responsible to create the object of servlet?

The web container or servlet container.

5) When servlet object is created?

At the time of first request.

7) What is difference between PrintWriter and ServletOutputStream?

PrintWriter is a character-stream class where as ServletOutputStream is a byte-stream class. The PrintWriter class can be used to write only character-based information whereas ServletOutputStream class can be used to write primitive values as well as character-based information.

8) What is difference between GenericServlet and HttpServlet?

The GenericServlet is protocol independent whereas HttpServlet is HTTP protocol specific. HttpServlet provides additional functionalities such as state management etc.

9) What is servlet collaboration?

When one servlet communicates to another servlet, it is known as servlet collaboration. There are many ways of servlet collaboration:

* RequestDispacher interface
* sendRedirect() method etc.

11) Can you call a jsp from the servlet?

Yes, one of the way is RequestDispatcher interface for example:

1. RequestDispatcher rd=request.getRequestDispatcher("/login.jsp");
2. rd.forward(request,response);

16) What is difference between Cookies and HttpSession?

Cookie works at client side whereas HttpSession works at server side.

18) How can we perform any action at the time of deploying the project?

By the help of ServletContextListener interface.

19) What is the disadvantage of cookies?

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

20) How can we upload the file to the server using servlet?

One of the way is by MultipartRequest class provided by third party.

23) What is war file?

A war (web archive) file specifies the web elements. A servlet or jsp project can be converted into a war file. Moving one servlet project from one place to another will be fast as it is combined into a single file.

24) How to create war file?

The war file can be created using jar tool found in jdk/bin directory. If you are using Eclipse or Netbeans IDE, you can export your project as a war file.

To create war file from console, you can write following code.

1. jar -cvf abc.war \*

Now all the files of current directory will be converted into abc.war file.

25) What are the annotations used in Servlet 3?

There are mainly 3 annotations used for the servlet.

1. @WebServlet : for servlet class.
2. @WebListener : for listener class.
3. @WebFilter : for filter class.

26) Which event is fired at the time of project deployment and undeployment?

ServletContextEvent.

27) Which event is fired at the time of session creation and destroy?

HttpSessionEvent.

28) Which event is fired at the time of setting, getting or removing attribute from application scope?

ServletContextAttributeEvent.

29) What is the use of welcome-file-list?

It is used to specify the welcome file for the project.

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.

1. 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.

1. 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.

1. 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.

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.

1. 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.

1. 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.

1. 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.

1. What is the use of servlet wrapper classes?
2. 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.
3. 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, thats why this interface is Deprecated in Servlet 2.4.

1. 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.

1. Is it good idea to create servlet constructor?

We can define a constructor for servlet but I don’t think its 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.

1. 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

1. 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.

1. 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.

1. 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.

1. What are life cycle methods of a servlet?

Servlet Life Cycle consists of three methods:

* + 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.
  + 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.
  + public void destroy() – This method is invoked once when servlet is unloaded from memory.

1. why we should override only no-agrs 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 get’s 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.

1. What is URL Rewriting?

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.

1. 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.

1. 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.

1. Why do we have servlet filters?

Some common tasks that we can do with filters are:

* + Logging request parameters to log files.
  + Authentication and authorization 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.

1. 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.

1. 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.

1. 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>

1. What is a deployment descriptor?

Deployment descriptor is a configuration file for the web application and it’s name is web.xml and it resides in WEB-INF directory. Servlet container use 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.

1. 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 loads, 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.

1. 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())

1. 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()

1. 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.

1. How to get the IP address of client in servlet?

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

1. What are important features of Servlet 3?

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

* + **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](http://www.journaldev.com/1877/servlet-tutorial-java#servlet-3-annotations).

* + **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.
  + **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.
  + **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.

1. 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.

1. 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.

For what purpose destroy() method of a servlet is used?

This method gives your servlet a chance to close database connections, halt background threads, write cookie lists or hit counts to disk, and perform other such cleanup activities.

For what purpose doGet() method of a servlet is used?

This method should be used to get data from server.

For what purpose doPost() method of a servlet is used?

This method should be used to process data on the server.

How to read form data in servlet?

Servlets handles form data parsing automatically using the following methods depending on the situation:

* **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.

What is HTTPServletRequest class?

When a browser requests for a web page, it sends lot of information to the web server which can not be read directly because this information travel as a part of header of HTTP request. HTTPServletRequest represents this HTTP Request.

What is HTTPServletResponse class?

when a Web server responds to a HTTP request to the browser, the response typically consists of a status line, some response headers, a blank line, and the document. HTTPServletResponse represents this HTTP Response.

How to write html contents using servlets?

Get the object of PrintWriter using request.

PrintWriter out = response.getWriter();

Now print html.

out.println("Hello World");

How to send an authentication error from a servlet?

We can use setStatus(statuscode) method of HttpServletResponse to send an authentication error.

// Set error code and reason.

response.sendError(407, "Need authentication!!!" );

How to redirect a request from a servlet to another servlet?

Page redirection is generally used when a document moves to a new location and we need to send the client to this new location or may be because of load balancing, or for simple randomization. The simplest way of redirecting a request to another page is using method sendRedirect() of response object.

How sendRedirect method works?

This method generates a 302 response along with a Location header giving the URL of the new document.

How sendError method works?

This method sends a status code (usually 404) along with a short message that is automatically formatted inside an HTML document and sent to the client.

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.

Can multiple filters be configured?

Yes.

Can filtering be done in an ordered way? If so then how to achieve it?

Yes. The order of filter-mapping elements in web.xml determines the order in which the web container applies the filter to the servlet. To reverse the order of the filter, you just need to reverse the filter-mapping elements in the web.xml file.

How to configure a central error handling page in servlets?

Use the error-page element in web.xml to specify the invocation of servlets in response to certain exceptions or HTTP status codes.

How to configure a central error handler in servlets?

If you want to have a generic Error Handler for all the exceptions then you should define following error-page instead of defining separate error-page elements for every exception:

<error-page>

<exception-type>java.lang.Throwable</exception-type >

<location>/ErrorHandler</location>

</error-page>

How to delete a session in servlet?

When you are done with a user's session data, you have several options:

* **Remove a particular attribute:** You can call public void removeAttribute(String name) method to delete the value associated with a particular key.
* **Delete the whole session:** You can call public void invalidate() method to discard an entire session. Setting Session timeout: You can call public void setMaxInactiveInterval(int interval) method to set the timeout for a session individually.

**Log the user out:** The servers that support servlets 2.4, you can call logout to log the client out of the Web server and invalidate all sessions belonging to all the users.

What is internalization?

This means enabling a web site to provide different versions of content translated into the visitor's language or nationality.

What is localization?

This means adding resources to a web site to adapt it to a particular geographical or cultural region for example Hindi translation to a web site.

What is locale?

This is a particular cultural or geographical region. It is usually referred to as a language symbol followed by a country symbol which is separated by an underscore. For example "en\_US" represents english locale for US.

How to detect locale in Servlets?

Following is the method of request object which returns Locale object.

java.util.Locale request.getLocale()

How to get country name in Servlets?

Following method returns a name for the locale's country that is appropriate for display to the user.

String getDisplayCountry()