**Servlet**

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Servlet is a technology i.e. used to create web application.

Servlet is an API that provides many interfaces and classes including documentations.

Servlet is an interface that must be implemented for creating any servlet.

Servlet is a class that extend the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.

Servlet is a web component that is deployed on the server to create dynamic web page.

CGI(Commmon Gateway Interface)

CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request. For each request, it starts a new process.

Disadvantages of CGI

There are many problems in CGI technology:

1. If number of clients increases, it takes more time for sending response.
2. For each request, it starts a process and Web server is limited to start processes.
3. It uses platform dependent language e.g. C, C++, perl.

Servlet Tutorial



**Servlet** technology is used to create web application (resides at server side and generates dynamic web page).

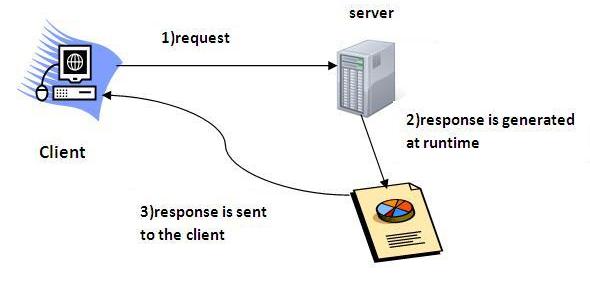
**Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was popular as a server-side programming language. But there was many disadvantages of this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse etc.

What is a Servlet?

Servlet can be described in many ways, depending on the context.

* Servlet is a technology i.e. used to create web application.
* Servlet is an API that provides many interfaces and classes including documentations.
* Servlet is an interface that must be implemented for creating any servlet.
* Servlet is a class that extend the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.
* Servlet is a web component that is deployed on the server to create dynamic web page.

   
  
Do You Know?

* What is the web application and what is the difference between Get and Post request ?
* What information is received by the web server if we request for a servlet ?
* How to run servlet in Eclipse, MyEclipse and Netbeans IDE ?
* What are the ways for servlet collaboration and what is the difference between RequestDispatcher and sendRedirect() method ?
* What is the difference between ServletConfig and ServletContext interface?
* How many ways we can maintain state of an user ? Which approach is mostly used in web development ?
* How to count total number of visitors and total response time for a request using Filter ?
* How to run servlet with annotation ?
* How to create registration form using Servlet and Oracle database ?
* How can we upload and download file from the server ?

What is web application?

A web application is an application accessible from the web. A web application is composed of web components like Servlet, JSP, Filter etc. and other components such as HTML. The web components typically execute in Web Server and respond to HTTP request.

CGI(Commmon Gateway Interface)

CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request. For each request, it starts a new process.

Disadvantages of CGI

There are many problems in CGI technology:

1. If number of clients increases, it takes more time for sending response.
2. For each request, it starts a process and Web server is limited to start processes.
3. It uses platform dependent language e.g. C, C++, perl.

Advantage of Servlet

There are many advantages of servlet over CGI. The web container creates threads for handling the multiple requests to the servlet. Threads have a lot of benefits over the Processes such as they share a common memory area, lightweight, cost of communication between the threads are low. The basic benefits of servlet are as follows:

1. **better performance:** because it creates a thread for each request not process.
2. **Portability:** because it uses java language.
3. **Robust:** Servlets are managed by JVM so we don't need to worry about memory leak, garbage collection etc.
4. **Secure:** because it uses java language..

**Servlet API**

**Servlet API** provides Classes and Interface to develop web based applications.

**Package**

Servlet API contains two java packages are used to developed the servlet programs, they are:

* javax.servlet
* javax.servlet.http

**javax.servlet**

**javax.servlet**package contains list of interfaces and classes that are used by the servlet or web container. These classes and interface are not specific to any protocol.

**javax.servlet.http**

**javax.servlet.http**package contains list of classes and interfaces to define http servlet programs. This package are used to interact with browser using http protocol. It is only responsible for http requests.

**Interfaces in javax.servlet package**

* Servlet
* ServletRequest
* ServletResponse
* RequestDispatcher
* ServletConfig
* ServletContext
* SingleThreadModel
* Filter
* FilterConfig
* FilterChain
* ServletRequestListener
* ServletRequestAttributeListener
* ServletContextListener
* ServletContextAttributeListener

**Classes in javax.servlet package**

* GenericServlet
* ServletInputStream
* ServletOutputStream
* ServletRequestWrapper
* ServletResponseWrapper
* ServletRequestEvent
* ServletContextEvent
* ServletRequestAttributeEvent
* ServletContextAttributeEvent
* ServletException
* UnavailableException

**Interfaces in javax.servlet.http package**

* HttpServletRequest
* HttpServletResponse
* HttpSession
* HttpSessionListener
* HttpSessionAttributeListener
* HttpSessionBindingListener
* HttpSessionActivationListener
* HttpSessionContext (deprecated now)

**Classes in javax.servlet.http package**

* HttpServlet
* Cookie
* HttpServletRequestWrapper
* HttpServletResponseWrapper
* HttpSessionEvent
* HttpSessionBindingEvent
* HttpUtils (deprecated now)

**Deployement Discriptor:**

1. Describes the purpose , semantics and correct structure of the following deployment descriptor elements.
   1. error-page
   2. init-param
   3. mime-mapping
   4. servlet
   5. servlet-mapping
   6. servlet-class
   7. servlet-name
   8. welcome-file
2. For every web application , we have to maintain one xml file named with web.xml
3. It should be placed inside WEB-INF directory and it is also known as Deployment Descriptor .
4. Web container will use this xml file to get web applications deployment information i.e., web.xml acts as a guide to container.
5. web.xml also provides declarative mechanism for customizing web application without touching source code ( to change userName , password , Database url's ). not required to touch Servlet source code , we can perform in web.xml

Anotomy of web.xml

* <web-app>
  1. <description>
  2. <display-name>
  3. <context-param>
  4. <servlet>
  5. <servlet-mapping>
  6. <error-page>
  7. <welcome-file-list>
  8. <filter>
  9. <filter-mapping>
  10. <listener>
  11. <session-config>
  12. <jsp-config>
  13. <security-constraint>
  14. <security-role>
  15. <auth-constraint>
  16. <mime-mapping>

</web-app>