**HISTORY OF JAVA WEB SERVLET and JavaServer Pages (JSP)**

* Java is built naturally with internet functionality. In **June 1997**, Sun Microsystems announced the Java Servlet.
* Java Servlet is one of the fundamental building block in developing different components of a Java Web specifically mainstream server-side Java which runs in a single process by using grain threads.
* While in **1998,** Sun released the JavaServer Pages **(JSP).**
* JSP is used to easily code a dynamic content of the web’s HTML pages.
* Both Servlets and JSPs lets the programmers build or develop a portable, easy to maintain, modular and scalable Web applications.

**THE JAVA SERVLET**

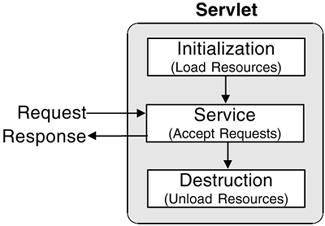
* In web applications, servlets were the first one which got the full access and power of Java. Wherein just like applets, it is completely and fully written in Java programming language.
* It is used for extending the server’s capability to host different applications that are accessed through a programming model, the request-response model. Which means it takes the request of the client and generates the response to the request.
* Java Servlet’s version 2.4 (current) which is included in the Java 2 Enterprise Edition (J2EE). Through the Tomcat project, it is available for free and also an open source.

**Features of Servlet 2.4:**

* Web Applications: In this section, servlet is always part of it (the web application), wherein it provides all the resources of a website.

**Servlet Container** – it manages all the Servlets in the basis of Web application.

*The web container facilitates the conversion to and from the HTTP request/response message to HTTP Servlet Request/HTTP Servlet Response*.

* Servlets and HTTP Servlets: With the support of HTTP, Servlet is used to provide a dynamic web pages and also plays an essential mechanism in Client/Server model.
* Filters: Used in authentication, logging and compression and it is an abstracted method that manipulates the request and response of the client before the request ends.
* Security: Servlets used the security that has been provided by JVM but also provides a way to control access of resources in the web application.
* Internalization: One of the best feature of a Servlet is that it can be develop content using variety of different languages by means of Servlet API.
* Servlet Life Cycle

1. *Instantiation:*

* The web container created the instance only once in the cycle.

1. *Initialization:* **init() method is invoked**

* This phase represents the creation of different resources to service requests. The init() method is invoked only once and before servicing of request takes place, the servlet invokes it first.

Syntax:

public void init(ServletConfig config) throws ServletException

1. *Request Handling:* **service() method is invoked**

* This phase represents all requests interactions invoked by each client.

Two parameters of service() method: Represents client’s request and Servlets’s client’s response.

1. javax.servlet.ServletRequest
2. javax.servlet.ServletResponse

Syntax:

public void service(ServletRequest request, ServletResponse response)

  throws ServletException, IOException

1. *Destruction:* **destroy() method is invoked**

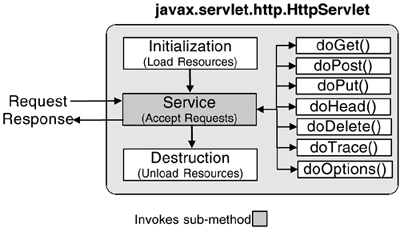
* Represents when the Servlet is removed in a container, simply, the destruction of a life cycle phase. The container calls the method destry() and terminate the resources that have been created.

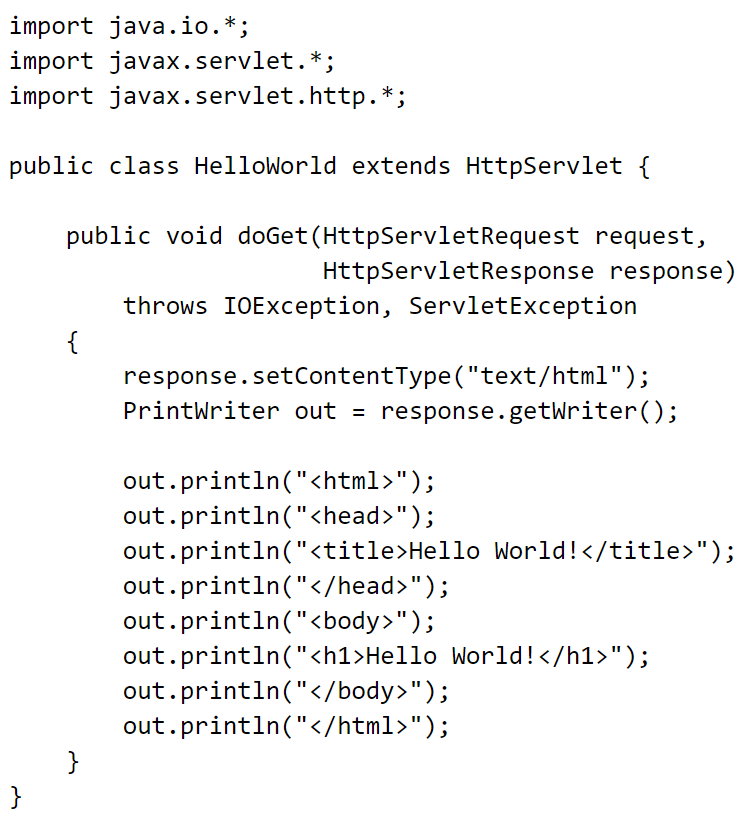
Syntax:

public void destroy()

* The HttpServlet object also uses the Servlet Life Cycle yet there are some modifications when it comes to HTTP protocol.
* During the phase of the service request, one of the seven helper methods is called which is appropriate to the type of HTTP request, named as follows:

1. doGet()
2. doPost()
3. doPut()
4. doHead()
5. doOptions()
6. doDelete()
7. doTrace()

The HttpServlet Life Cycle

* The figure below is a sample code for HttpServlet which will generate a simple HTML page.
* When deploying a servlet, it is not the case that it is written fully in java. For the client to access the Servlet, a unique URL is declared within the Web Application Deployment Descriptor which is the web.xml.
* The web.xml relies on the following two new elements:

1. servlet element – loaded by the web application to define the a Servlet.
2. Sevlet-mapping element – map the Servlet in a URL with the following servlet specification:
3. **An exact pattern to match.** The pattern must start with a /, but can contain anything afterwards. This type of pattern is used for a one-to-one mapping of a request to a specific Servlet.
4. **An extension match, \*.extension.** In this case all URLs ending with the given extension are forwarded to the specified Servlet. This is commonly used in Servlet frameworks and can force many requests to go to the same Servlet[15](javascript:popUp('/content/images/chap2_0321136497/elementLinks/ch02fn15.html')).
5. **A path mapping.** Path mappings must start with a / and end with a /\*. In between anything can appear. Path mappings are usually used for forwarding all requests that fall in a certain directory to a specific Servlet.
6. **Default Servlet, /.** A default Servlet mapping is used to define a Servlet for forwarding requests when no path information is given. This is analogous to a directory listing.