**SERVER-SIDE WEB SCRIPTING**

**Java Servlets**

* This are Java objects that are intended to components in client server communications.
* It handles clients’ requests by generating responses to such requests.
* It is a part of the JAVA EE specification (see java servlet API for more details)
* Hosted in a ‘***servlet container***’ that provides the environment in which the servlet runs, as well as controls the ‘***servlet life cycle***’.

|  |
| --- |
| **NOTE:**  ***‘Java Servlet’*** is an object that hosted in ‘servlet container’.  ***‘Servlet Container’*** is an environment. |

**Servlet Lifecycle**

* **instantiation**  -> constructor call

- creates an instance of the servlet service client requests.

- deployment of the descriptor (***web.xml***) for the web application.

- invoked implicitly by the server container when the servlet is called upon to service a client request & no instance currently exists.

* **initialization**  -> int() method

- invoked only once & is intended for any startup initialization code for servlet.

- an action will occur in parsing the context of the application (***web.xml***).

- ***Example of specifying an element in web.xml:***

<servlet>

<servlet-name>

WebTechServletDemo

</servlet-name>

<servlet-class>

com.coursenotes.WebTechServletDemo

</servlet-class>

<load-on-startup>

1

</load-on-startup>

</servlet>

\*This method is called once in this server life cycle. The ***com.coursenotes.WebTechServletDemo*** is the corresponding method to the server class, and this will invoke the web container.

* **request handling**  -> service() method

- this method handle the request being send to the server.

- different types of request:

* POST
* GET
* HEAD
* PUT
* DELETE

- invoked for each client request are typically handled by a single, multi-threaded servlet instance; thus, must be exercised to ensure that the code being run by servlet is ‘safe thread’

- **destruction** -> destroy() method

- invoke for shutting down or

- invoked only once before unloading the servlet.

**HTTP Servlet**

* Java servlet is used to handle http request & generate http responses.
* This servlet is tied to HTTP Protocol.
* This servlet is used to implement an application’s web layer.
* HTTP Servlet is dependent and used specifically to a HTTP protocol.
* Hosted in ‘***web container’*** which a component in java ‘***application server***’ (e.g. Apache TomCat, Red Hat JBoss / WildFly, Oracle GlassFish, IBM Web Sphere)
* The service() method call is routed to a doXXX() call, depending on the http request method
  + doGet() }
  + doPost()
  + doHead()
  + doPut()
  + doDelete()
  + doOptions()
  + doTrace()

***EXAMPLE ON HANDLING BOTH GET & POST REQUEST:***

public class DemoHttpServlet extends HttpServlet{

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException{

response.getWriter().write(“<html><body>GET response</body></html>”);

}

* ***The doXXX() methods are passed in 2 arguments:***
  + **HTTPServletRequest**
    - Object representation of the HTTP Request sent by the client & received by the servlet.
    - Used to access information from the request message
      * HTTP Request Method
      * Request URI
      * Query String
      * Message Readers
      * Message Payload
      * etc.
      * **HTTPServletResponse**
      * The object representation of the HTTP response generated by the servlet & send back to the requesting client.