virtusa®

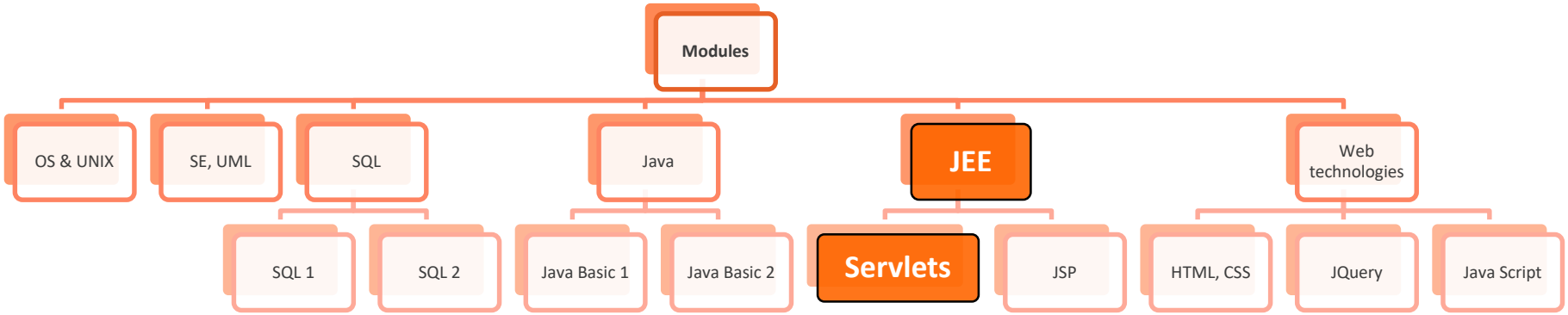
Accelerating Business Outcomes

Java Servlets

Module Overview

Purpose:

The following modules are identified to build the basic IT skills and acquaint you with the technology basics. The current module (highlight in red) is contributing on Basics of Java Programming.



* **Recommended duration**: 10 **hours**

Module Objectives

By the end of this module, you will be able to:

- Understand Servlets and its Architecture
- Servlets Life Cycle
- Servlets Chaining
- Servlet Session Tracking

Introduction to Servlets

Java Servlet

- Servlet are java classes that extends the functionality of a web server by dynamically generating web pages. A run time environment is known as servlet container
- Manages servlet loading and unloading and work with the web server to direct request to the servlet and to send output back to web client.

The Key advantage of the servlet

- Performance : Servlet by contrast loaded in memory when first requested, and stay in memory indefinitely.
- Simplicity-:It require basic HTTP to communicate with their clients. No special client side software is required, even with older browser.
- Advantage of servlet Over traditional CGI
- Http session-: Although HTTP is stateless protocol. Servlet API provides Http session class that over comes this limitations.

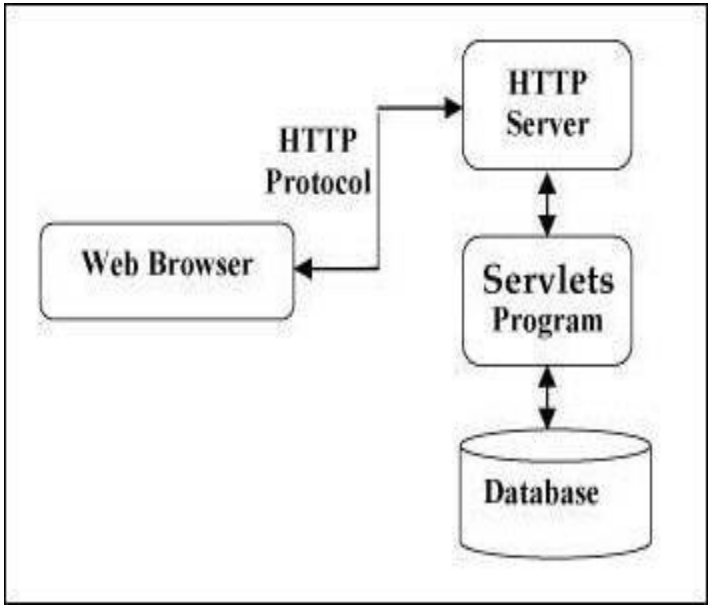
Access to java Technology

- It has a full privilege to access the java technology. Such as thread networking access and database connectivity.

References	https://www.tutorialspoint.com/servlets/ https://www.journaldev.com/1877/servlet-tutorial-java

Servlets Architecture

- Servlet is a class, which implements the javax.servlet.Servlet interface. However instead of directly implementing the javax.servlet.Servlet interface we extend a class that has implemented the interface like javax.servlet.GenericServlet or javax.servlet.http.HttpServlet.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.



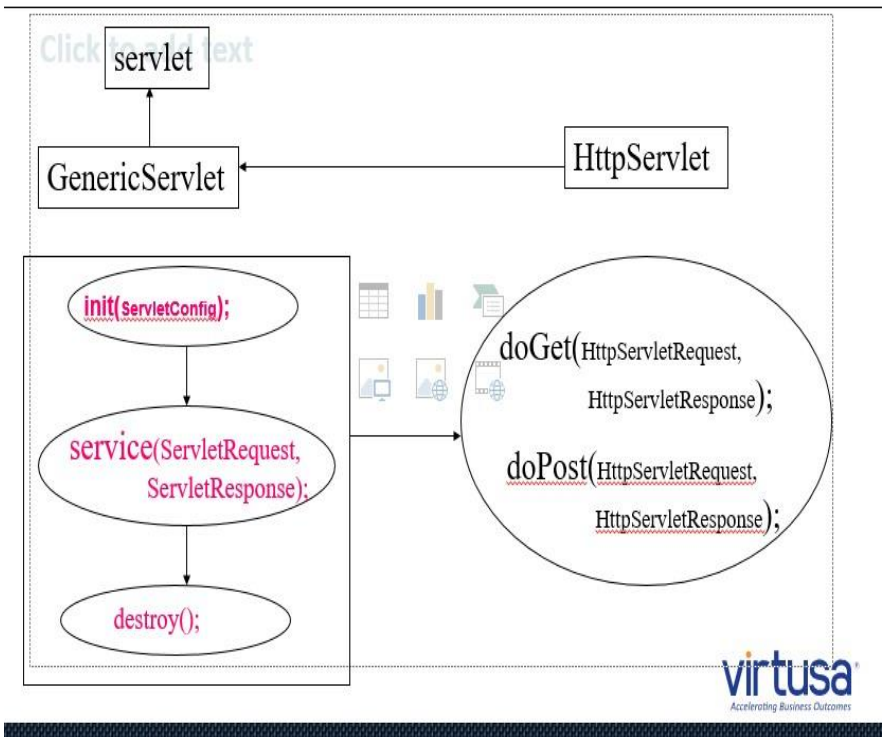
References

- https://www.tutorialspoint.com/servlets/servlets_overview.htm

Servlets Life Cycle

Servlet life cycle is similar with the applet life cycle
For Example-:
It provides the method such as init(), start(), paint(), stop(), destroy() which are called by applet runtime environment in response to user request .
Similarly servlet operate in the context of a request and response model managed by servlet Container.

It does the following steps-:
Load the servlet when it is first requested Call the servlet's init() method.
Handel the request by calling the service() method.
Call destroy() method of each active servlets.



References	https://www.tutorialspoint.com/servlets/servlets-life-cycle.htm

Servlets Continued..

Fetch Form Data using 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 ex checkbox.
- getParameterNames() – Call this method if you want a complete list of all parameters in the current request
- Syntax:

```
String firstname =request.getParameter("first_name")
```

References	https://www.tutorialspoint.com/servlets/servlets-form-data.htm

Servlets Continued..

Fetch Form Data using 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 ex checkbox.
- getParameterNames() − Call this method if you want a complete list of all parameters in the current request
- Syntax:

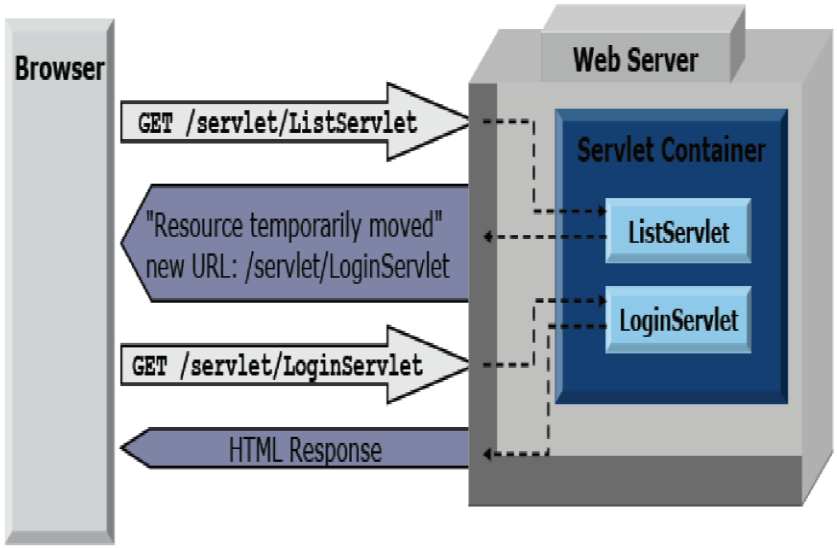
```
String firstname =request.getParameter("first_name")
```

Reference	https://www.tutorialspoint.com/servlets/servlets-life-cycle.htm

Servlet Chaining

Redirect

- Redirects the browser to request a different URL
- Browser sends a completely new Request
- The original request parameters are not included HTML



- https://www.javatpoint.com/sendRedirect()-method

Redirect Example

```
if(!validateCustomer("CustomerID") )
{
response.sendRedirect(response.encodeRedirectUrl
("LoginServlet?Origin=SharePortfolio.jsp"));

else{
// do thereal thing
```

Forward Example

- Assigns the same Request to a different Servlet
- Internal to the servlet engine
- Request object is used as-is (including all parameters)
- Forward is invisible to browser

https://www.javatpoint.com/requestdispatcher-in-servlet

Session Tracking

Http is a stateless protocol which does not maintain any of the user information across the pages.

Following techniques will help us to maintain the session

Hidden <form> fields can be used to store a unique ID for the session

Cookies are small files that the servlet can store on the client computer, and retrieve later

URL rewriting: You can append a unique ID after the URL to identify the user

Java's Session (HttpSession) Tracking API can be used to do most of the work for you

Session Object

To get the user's session object, call the getSession() method of the HttpServletRequest class.

Example: HttpSession session = request.getSession();

Reference	https://www.javatpoint.com/session-tracking-in-servlets

Session Tracking cont.

If user already has a session, the existing session is returned. If no session exists, a new one is created and returned.

If you want to know if this is a new session, call the Session isNew() method.

Example:

boolean isNew() return true if the client hasn't yet joined the session. This is true when the session is first created and session ID is passed to the client.

Encoding URLs

If a browser does not support cookies, you need some other way to maintain the user's session ID.

The Servlet API takes care of this for you by automatically appending the session ID to URLs if the browser does not support cookies.

To automatically append the session ID, use the encodeURL () method } String url = response.encodeURL (originalURL);

Hidden fields

If the application consist of a series of application forms using submit button for navigation, the session ID can be stored as a hidden field. That is retrieved with request.getParameter()

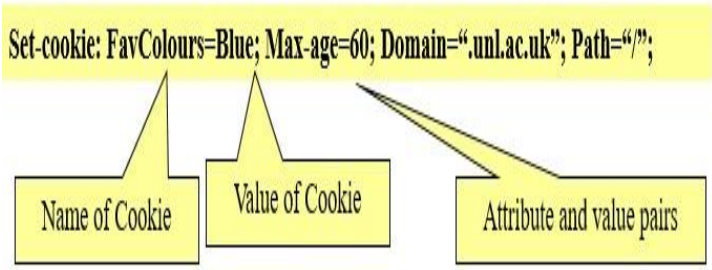
Example-:

```
<input type="hidden" name="sessionID" value="...">
```

Servlets

Cookies-: The server can send a set-Cookie header in it's initial response with the session ID as the value of the cookie. On the subsequent request, the client can return the value with a cookie header.

- Originally invented by Netscape are exchanged in header fields of request and response information of the HTTP messages. One advantage with cookies is that it does not involve modifying dynamically generated URLs or forms. It is possible that some users may choose to disable cookies; in this case another strategy must be chosen.
- Cookies are the most popular way to support sessions. Web servers send cookies to the client in response header



- https://www.javatpoint.com/session-tracking-in-servlets

Additional References

The last discussed topics aimed at to gain the concept knowledge and the practical contexts to apply the knowledge.

If you want to explore more and build the expertise level, refer to below links and books:

Links :-

https://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html

Books :-

- **Head First Servlets and JSP**

Self Check?

Instructions to write Self Evaluation Sheet:

Open the excel sheet, refer Servlets sheet, write down the solutions for all questions, save a local copy in your machine.

Lab Assignment

- Refer Assignment Document to complete the tasks on the required timeline
- You are required to submit the Solutions for the given assignment and refer the Participant guide to get know the submission procedure.

Module Summary

Now that you have completed this module, you will be able to:

- Develop JSP using different building blocks
- Understand Scripting elements, implicit objects and directives
- Use standard action tags and custom tags
- Use JSTL library with Jsp

Thank you!