



# Servlet

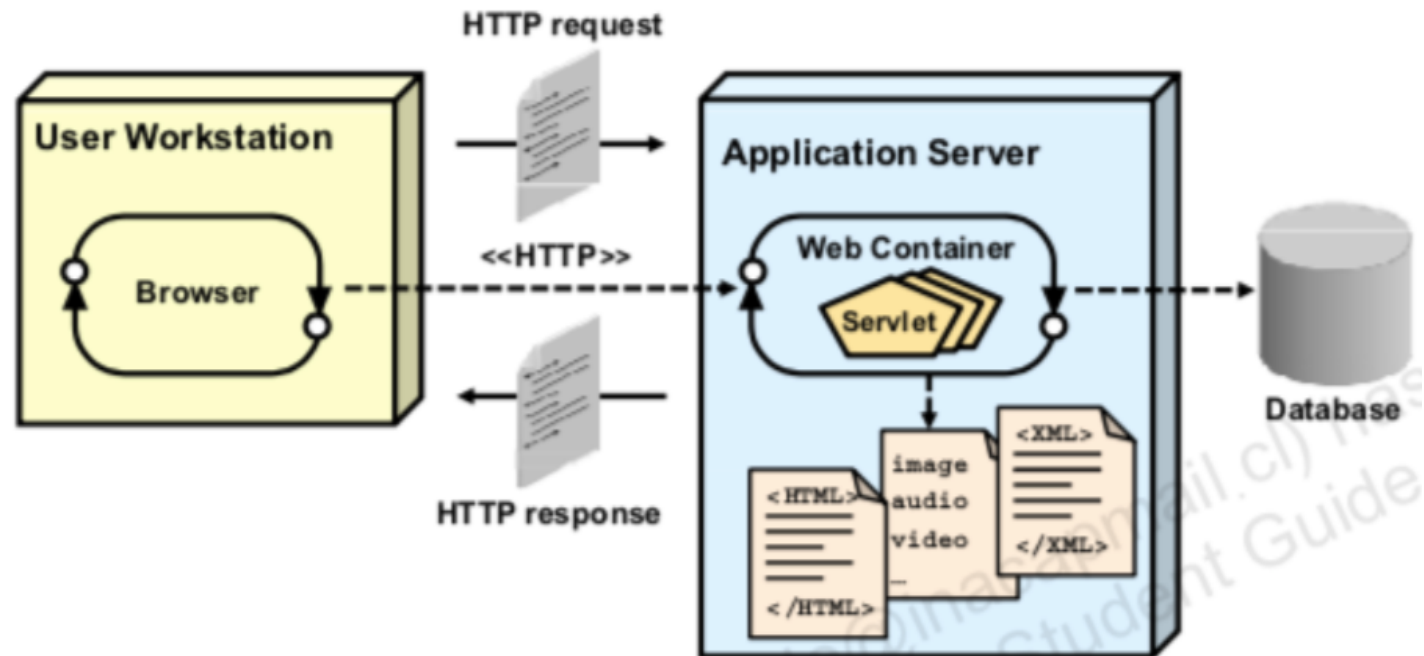
Características

- Es un componente de Java Web, específicamente es una Java Class.
- Recibe el nombre de Controlador o Controller
- La función del controlador es procesar las peticiones enviadas por un formulario o un link , y entregar una respuesta.



# Servlet

Esquema





# Servlet

Código

```
1 package controller;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.*;
7
8 @WebServlet("/control1")
9 //more names
10 //@WebServlet(urlPatterns={"/control1", "/setup"})
11 public class Control1 extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14
15     protected void doGet(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17
18     }
19
20
21     protected void doPost(HttpServletRequest request, HttpServletResponse response)
22         throws ServletException, IOException {
23
24     }
25
26 }
```



# Servlet

## Características

- Los servlet poseen un nombre físico y un nombre lógico cuya propiedad se denomina url-pattern y puede ser definida al momento de crear al servlet.
- Posee dos métodos denominados **doGet** para peticiones GET y **doPost** para peticiones POST que estan ligadas al uso de formularios.
- Ambos métodos poseen dos objetos que son utilizados para el manejo de las peticiones y respuestas a un formulario y son denominadas **HttpServletRequest** y **HttpServletResponse**.



# Servlet

HttpServletRequest

- Es una interfaz la cual posee algunos métodos dentro de los cuales se destacan
  - **getParameter(String param):** permite capturar el valor de un control de formulario, retorna siempre un String.
  - **getParameterValues(String param):** permite capturar un conjunto de datos especialmente cuando se trabaja con checkbos, retorna un arreglo de String.
  - **setAttribute(String name, Object object):** permite compartir un recurso a otro componente web.
  - **getAttribute(String name):** permite capturar un recurso compartido, retorna siempre un Object
  - **getRequestDispatcher(String url):** permite redireccionar al componente web al cual se quiere compartir un recurso.
  - **getSession():** retorna un objeto de session.



# Servlet

HttpServletResponse

- Es una interfaz la cual posee algunos métodos dentro de los cuales se destacan
  - **getWriter()**: retorna un objeto `PrintWriter` que permite que un servlet pueda generar una salida HTML
  - **sendRedirect(String url)**: redirecciona a una url
  - **addCookie(Cookie cookie)**: agrega una `Cookie` al navegador.



# JSP

Java Server Page

- Es un componente que actua como una vista.
- Es muy similiar a un documento HTML.
- La gran diferencia es que en un JSP se puede incrustar código Java.



# JSP

Java Server Page

- La incrustación de código recibe el nombre de scriptlet y pueden ser:
  - `<% %>`: para incrustar código Java
  - `<%= %>`: para mostrar una variable Java en en browser
  - `<%@page%>`: para importar dependencias Java.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
  </head>
  <body>
    <h3>JSP demo</h3>
    <% for(int i=1; i<=10; i++) { %>
      <p> parrafo <%= i %></p>
    <% } %>
  </body>
</html>
```



# PRACTICO 1

- En el body de un JSP muestre los doce meses utilizando un for de java. Ejemplo de salida

## Práctico 1

Los doce meses son

- Mes 1
- Mes 2
- Mes 3
- Mes 4
- Mes 5
- Mes 6
- Mes 7
- Mes 8
- Mes 9
- Mes 10
- Mes 11
- Mes 12

## PRACTICO 2

- Crea un arreglo con frutas, recorre el arreglo y muestra las frutas como en el ejemplo:
  - `String []frutas = {"Guinda", "Uva", "Sandía", "Piña"}`

### Práctico 2

Mis frutas

1. Guinda
2. Uva
3. Sandía
4. Piña



# EL

Expression Language

- El objetivo de EL es poder simplificar la escritura de Java en un código HTML.
- Trabaja mediante una notación denominada `${code}`, por lo tanto, es un componente que está relacionado con las variables u objetos definidos en un proyecto.
- Donde se debe conocer su sintaxis y propiedades.
- Por ejemplo los operadores lógicos pueden ser:
  - and ó &&
  - or ó ||
  - not ó !
  - empty ó null



# JSTL

JSP Standard Tag Library

- Es una colección que se trabaja por medio de tags, con el objetivo de poder encapsular la funcionalidad de Java en tags fáciles de leer y utilizar
- Los JSTL dependiendo del AS (Application Server), se deben instalar.
- Se clasifican en
  - Core Tags
  - Formatting tags
  - SQL tags
  - XML tags
  - JSTL functions



## JSTL CORE TAGS

- Se debe importar en el documento JSP
  - `<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>`
- Sus tags mas reelevantes son:
  - `<c:set>`
  - `<c:out>`
  - `<c:if>`
  - `<c:forEach>`
  - `<c:import>`
  - `<c:choose>` | `<c:when>` | `<c:otherside>`
  - `<c:param>`
  - `<c:redirect>`
- Referencia:  
[https://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)