Introducción a Java Servlets y JSP

Sesión Experimental 1

{desafío} latam_

- Activación de conceptos clave
- Desafío
- Panel de discusión

Inicio

15 minutos

{desafío} latam_

/* Activación de conceptos */

Creación de un Servlet

Ya sabemos que un servlet es una clase java común y corriente, con ciertas características especiales. A continuación veremos los pasos a seguir para generar el primer servlet, y detallaremos la relación de herencia que encontramos en las clases involucradas.

Toda clase que usaremos como servlet, debe extender de la clase padre HttpServlet. Esto para heredar toda la funcionalidad que nos provee el api de Java EE.

```
@WebServlet("/processForm")
public class MyServlet extends HttpServlet {
```

Servlet Básico

Anotación que disponibiliza la llamada por web.

La clase hereda de HttpServlet.

Método doGet, encargado de procesar la lógica a implementar.

Parámetros de entrada, request y response.

Objeto PrintWriter para imprimir salida de texto html mediante el response.

Con objeto writer generamos dinámicamente el HTML.

```java
package com.desafiolatam.inicioservlet;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/saludo")
public class InicioServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        PrintWriter writer = response.getWriter();
        //comenzamos a generar el html
        writer.println("<html>");
            writer.println("<body>");
                writer.print("<h1>");
                writer.print("Hola, soy una web generada por un servlet");
                writer.print("</h1>");
            writer.println("</body>");
        writer.println("</html>");
    }
}
```

{desafío} latam_

Pasar valores a servlet

```java
package com.desafiolatam.inicioservlet;

import java.io.IOException;

@WebServlet("/saludo")
public class InicioServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {

        PrintWriter writer = response.getWriter();
        String nombre;
        String apellido;
        /*Rescatamos un nombre y un apellido enviados por url*/
        nombre = request.getParameter("nombre");
        apellido = request.getParameter("apellido");

        writer.println("<html>");
            writer.println("<body>");
                writer.print("<h1>");
                writer.print("Hola " + nombre + " " + apellido);
                writer.print("</h1>");
            writer.println("</body>");
        writer.println("</html>");
    }
}
```

Mediante el objeto request, es posible obtener los valores enviados por el usuario.

Redirigir a otro servlet

Con la sentencia getRequestDispatcher(“/webMethod”) puedes redirigir a otro servlet para que procese los datos.

```
request.getRequestDispatcher("/generaFactura").forward(request, response);
```

Desarrollo

150 minutos

{desafío} latam_

Cierre

15 minutos

{desafío} latam_

¿Existe algún concepto que no hayas comprendido?

Volvamos a revisar los conceptos que más te hayan costado antes de seguir adelante

Reflexionemos

{desafío} latam_

Futura implementación con base de datos

Según las técnicas de envio de parametros entre servlets, se puede notar que las capas se conectan mediantes referencias entre las direcciones de los servlets. En caso de tener que persistir los datos a una base de datos relacional

¿Cómo sería la integración con dicha capa de datos a nivel conceptual?

No importa la técnica o la sintaxis, la idea es imaginar la comunicación entre capas.

{desafío} latam_

Academia de talentos digitales