TABLA DE CONTENIDO

[TABLA DE CONTENIDO 1](#_Toc503636502)

[Crear Servlet 2](#_Toc503636503)

[Objetivo: 2](#_Toc503636504)

[Crear Servlet. 2](#_Toc503636505)

[Notas: 3](#_Toc503636506)

[Probar el servlet método GET 4](#_Toc503636507)

[Prueba intern de envio de mensaje desde java a la consola del navegador 4](#_Toc503636508)

[Probar el servlet método POST 5](#_Toc503636509)

[Teclear el país: Venezuela e introducir click en botón enviar. 5](#_Toc503636510)

[Leyenda 6](#_Toc503636511)

[Java EE Servlet y JSP 6](#_Toc503636512)

Crear Servlet

### Objetivo:

Crear un servlet.

### Crear Servlet.

1. Crear paquete: com.pais.servlet
2. Crear clase MiServlet que extienda de HTTPServlet.

**package com.pais.servlet;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.Servlet;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import org.osgi.service.component.annotations.Component;**

**@Component(**

**immediate = true,**

**property = {**

**"osgi.http.whiteboard.context.path=/",**

**"osgi.http.whiteboard.servlet.name=com.pais.servlet.MiServlet",**

**"osgi.http.whiteboard.servlet.pattern=/miservlet",**

**},**

**service = Servlet.class)**

**public class MiServlet extends HttpServlet {**

**private static final long serialVersionUID = -1970334471631842726L;**

**// Invocar http://localhost:8080/o/miservlet -> [Peticion GET]**

**@Override**

**protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {**

**System.out.println("Peticion doGet Servlet...!");**

**resp.setContentType("text/html");**

**PrintWriter out = resp.getWriter();**

**out.println("<html>");**

**out.println("<head>");**

**out.println("<title>Response Servlet GET</title>");**

**out.println("</head>");**

**out.println("<body>");**

**out.println("<h1>Respuesta .. peticion doGet Servlet...!!</h1>");**

**out.println("<br>");**

**out.println("<h2>Realizar una peticion POST</h1>");**

**out.println("<hr>");**

**out.println("<form action='miservlet' method='post'>");**

**out.println("<lb>");**

**out.println("<strong>Pais: </strong> <input name='pais' />");**

**out.println("<input type='submit' />");**

**out.println("</form>");**

**out.println("</body>");**

**out.println("</html>");**

**}**

**// Invocar http://localhost:8080/o/miservlet -> [Peticion POST]**

**@Override**

**protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {**

**System.out.println("Peticion doPost Servlet...!");**

**resp.setContentType("text/html");**

**String pais = req.getParameter("pais");**

**PrintWriter out = resp.getWriter();**

**out.println("<html>");**

**out.println("<head>");**

**out.println("<title>Response Servlet POST</title>");**

**out.println("</head>");**

**out.println("<body>");**

**out.println("<h1>Respuesta .. peticion doPOST Servlet...!!</h1>");**

**out.println("<hr>");**

**out.println("<lb>");**

**out.println("<strong>Pais:</strong>" + pais);**

**out.println("</body>");**

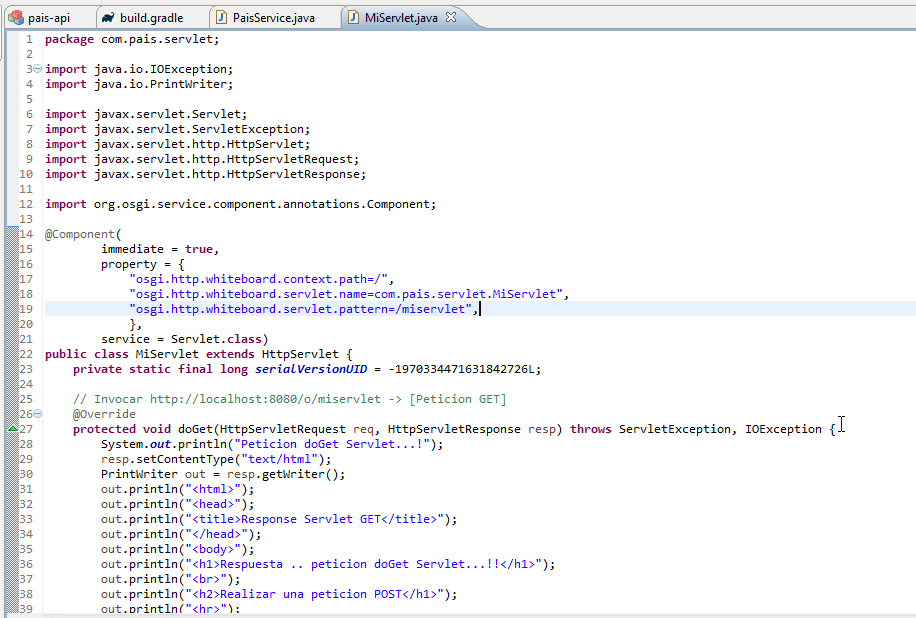
**out.println("</html>");**

**}**

**}**

### Notas:

* Como modulo OSGI con las propiedades:
  + **osgi.http.whiteboard.context.path=/**
  + **osgi.http.whiteboard.servlet.name=com.pais.servlet.MiServlet**
  + **osgi.http.whiteboard.servlet.pattern=/miservlet**
* Se genera los metodos GET y POST
  + GET: responde un Html (‘JSP’) con formulario para realizar una petición submit
  + POST: responde la petición post.



### Probar el servlet método GET

Con el servidor levantado. En el navegador invocar: <http://localhost:8080/o/miservlet>

Repuesta a petición GET:



### Prueba intern de envio de mensaje desde java a la consola del navegador

// Enviar un Alert de JavaScript como pruba

out.println("<script type=\"text/javascript\">");

out.println("console.log('Mensaje desde Java a la consola del navegador...!');");

out.println("</script>");

### Probar el servlet método POST

### Teclear el país: Venezuela e introducir click en botón enviar.

Repuesta a petición POST:

### 

Leyenda

### Java EE Servlet y JSP

**Servlet**

* Muy útiles para leer cabeceras de mensajes, datos de formularios, gestión de sesiones, procesar información, etc.
* Pero tediosos para generar todo el código HTML
* El mantenimiento del código HTML es difícil
* Los servlets tienen que generar todo el código HTML

**JSP (JavaServer Pages)**

* Fichero con código (X)HTML que incluye scripts codificados en Java
* Permite usar (X)HTML para definir gran parte de la página
* E introducir código Java en las partes dinámicas de la página
* Mediante etiquetas especializadas (Custom Tags) que amplían la sintaxis de HTML
* Se compila y se convierte en un servlet (solo la primera vez que se invoca)
* Se ejecuta como un servlet.
* Las JavaServer Pages (JSP) permiten escribir código HTML e insertar código Java para las partes dinámicas.

Con JSP es más fácil que se distribuya la tarea de diseño de la página web y la programación de la aplicación web

**Objetos predefinidos**

**Request**

* Objeto HttpServletRequest que permite acceder a la información de la solicitud

**Response**

* Objeto HttpServletResponse para generar la respuesta

**Sesión**

* Objeto HttpSession asociado a la petición
* Si no hubiera sesión será null

**Out**

* Objeto JspWriter (similar a un PrintWriter) para generar la salida para el cliente

**Application**

* El objeto ServletContext del contenedor web

**Objetos predefinidos**

La clase que va a ser nuestro servicio Web no es más que una clase normal de java, con métodos. Únicamente hay que poner las anotaciones específicas de *JAX-RS* para que el contenedor de aplicaciones lo convierta en servicio web cuando arranque. Las anotaciones son las siguientes:

**Objetos predefinidos**

La clase que va a ser nuestro servicio Web no es más que una clase normal de java, con métodos. Únicamente hay que poner las anotaciones específicas de *JAX-RS* para que el contenedor de aplicaciones lo convierta en servicio web cuando arranque. Las anotaciones son las siguientes: