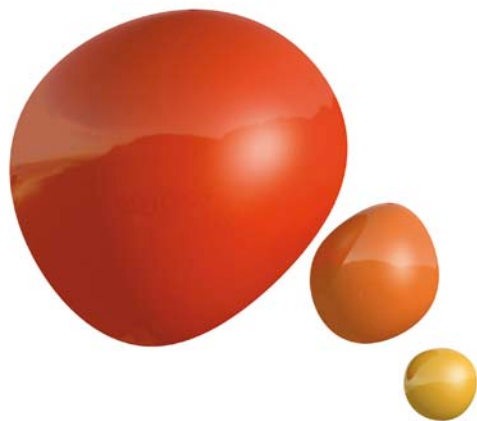




RECUPERACIÓN DE DATOS ENVIADOS EN LA PETICIÓN



ÍNDICE

RECUPERACIÓN DE DATOS ENVIADOS EN LA PETICIÓN

1. Envío de parámetros desde un formulario 3
2. Cabeceras de una petición 8



Recuperación de datos enviados en la petición

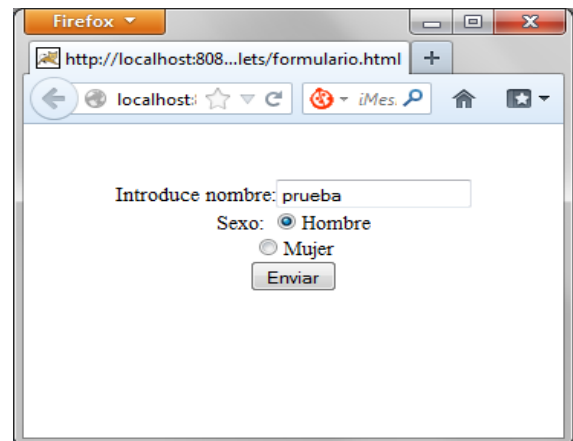
1. Envío de parámetros desde un formulario

La forma más habitual de enviar datos al Servlet desde una página Web es utilizando un formulario. Los datos introducidos por el usuario en los controles son enviados como parámetros al Servlet al producirse la petición.

```
<form action="procesado" method="post">
  Introduce nombre:<input name="myname" type="text"/><br/>
  Sexo: <input name="sexo" type="radio" value="h"/>Hombre<br/>
  <input name="sexo" type="radio" value="m"/>Mujer<br/>
  <input type="submit" value="Enviar"/>
</form>
```

Petición HTTP

	<code>myname=prueba&sexo=h</code>
--	---------------------------------------



“Los parámetros se envían como parejas nombre=valor, encadenando una pareja con la siguiente mediante un &. Los parámetros pueden enviarse en el cuerpo de la petición (petición POST) o en la cabecera pegados a la URL (petición GET)”.

Envío de parámetros desde enlaces

También se pueden enviar datos al Servlet a través de los enlaces que desde la página apuntan al mismo. En este caso, los datos no son introducidos explícitamente por el usuario; los parámetros son codificados directamente sobre la dirección que apunta al Servlet.

Recuperación de datos enviados en la petición

Pulse sobre el estilo preferido:

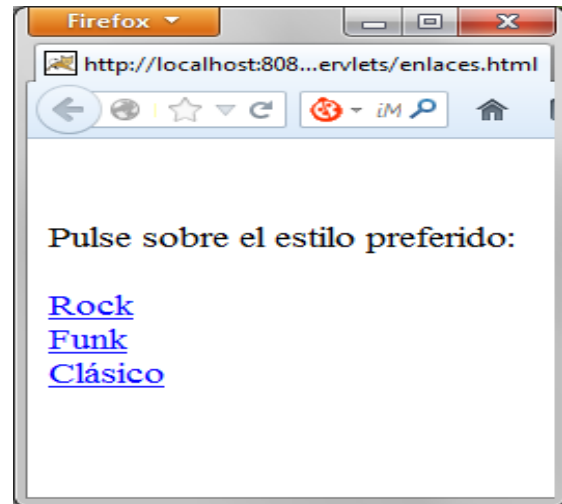
 Rock

 Funk

 Clásico

Petición HTTP

/procesado?tema=2	
-------------------	--



“Cuando los parámetros se anexan a la URL del Servlet, el tipo de envío es siempre GET”

La interfaz `HttpServletRequest`

Para recuperar desde el método de servicio los parámetros enviados desde una página al Servlet, disponemos de la interfaz `HttpServletRequest`.

El contenedor Web proporciona al método de servicio un objeto que implementa `HttpServletRequest`. A través de este podremos acceder tanto a los parámetros como a la información de cabecera generada por el navegador.

La interfaz `HttpServletRequest` tiene, entre otros, los siguientes métodos:

`String getParameter (String name)`

`String[] getParameterValues()`

`String getHeader(String name)`

Para recuperar desde el método de servicio los parámetros enviados desde una página al Servlet, disponemos de la interfaz `HttpServletRequest`.



Recuperación de datos enviados en la petición

Recuperación de valores de parámetros

Tanto si los datos son enviados mediante GET o como si los enviáramos mediante POST, la recogida se realizará con el método `getParameter (String name)`.

El valor de retorno es un String correspondiente al valor unicode de la primera ocurrencia del parámetro.

Se devuelve un String vacío si el parámetro existe pero no tiene valor, y se devuelve null si no existe dicho parámetro.

Si el parámetro pudiera tener más de un valor, deberíamos llamar a `getParameterValues ()` en vez de a `getParameter ()`. Este devuelve un array de strings.

```
protected void service(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        String msg;
        String sexo=request.getParameter("sexo");
        if (sexo.equals("h")){
            msg="sr."+request.getParameter("myname");
        }
        else{
            msg="sra."+request.getParameter("myname");
        }
        out.println("<html>");
        out.println("<body>");
        out.println("<h1>Bienvenido "+msg+"</h1>");
        out.println("</body>");
        out.println("</html>");

    } finally {
        out.close();
    }
}
```

Parámetros con múltiples valores

Si tenemos varios controles con el mismo nombre, a la hora de enviar los datos en la petición se enviará un único parámetro con múltiples valores:



Recuperación de datos enviados en la petición

```
<form action="procesado" method="post">
  Seleccione idiomas: <br/>
  <input name="idioma" type="checkbox" value="en"/>Inglés<br/>
  <input name="idioma" type="checkbox" value="es"/>Español<br/>
  <input type="submit" value="Enviar"/>
</form>
```

Para recuperar todos los idiomas seleccionados utilizaremos el método:

String [] getParameterValues(String name). Devuelve un array con todos los valores del parámetro.

Ejemplo

```
//genera una página que muestra los códigos de
//todos los idiomas elegidos
String[] idiomas=request.getParameterValues();
if(idiomas!=null){
    for(String c:idiomas){
        out.println("Código de idioma: "+c);
    }
}
```

Recogida de todos los parámetros

Para recuperar todos los parámetros que se reciben en una petición podemos recurrir al siguiente método del objeto `HttpServletRequest`:

Enumeration getParameterNames(). Devuelve una enumeración con los nombres de todos los parámetros recibidos en la petición.

Recuperación de datos enviados en la petición

```
<HTML>
<HEAD>
<TITLE> Formulario de Registro </TITLE>
</HEAD>
<BODY>
<FORM method="post" action="/servlet/victor.RegistroCliente">
Nombre:
<INPUT type="text" name="nombre" size="40"> <BR>
Direccion:
<INPUT type="text" name="direccion" size="50"> <BR>
<P>
Código Postal:
<INPUT type="text" name="codigopostal" size="10"
length="5"> <BR>
Provincia:
<SELECT name="provincia">
<OPTION selected value="madrid"> Madrid </OPTION>
<OPTION selected value="barcelona"> Barcelona </OPTION>
<OPTION selected value="coruna"> La Coruña </OPTION>
<OPTION selected value="ciudadreal"> Ciudad Real </OPTION>
<OPTION selected value="valencia"> Valencia </OPTION>
</SELECT> </P>
<P>
Teléfono de Contacto:
<INPUT type="text" name="telefono" size="15">
Email:
<INPUT type="text" name="email" size="30">
</P>
<P>
Sexo:
<INPUT type="radio" name="sexo" value="varon" checked> Varón
<INPUT type="radio" name="sexo" value="mujer"> Mujer
</P>
<P>
<INPUT type="submit" value="Grabar">
<INPUT type="reset" value="Eliminar">
</P>
</FORM>
</BODY>
</HTML>
```

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class RegistroCliente extends HttpServlet {
    public void doPost (HttpServletRequest petición,
        HttpServletResponse respuesta)
        throws ServletException, IOException
    {
        StringBuffer parametros = new StringBuffer();
        Enumeration e = petición.getParameterNames();
        while (e.hasMoreElements()) {
            String parametro = (String) e.nextElement();
            parametros.append (parametro);
            parametros.append ("=");
            parametros.append (petición.getParameter(parametro));
            parametros.append ("<BR>");
        }
        respuesta.setContentType("text/html");
        PrintWriter out = respuesta.getWriter();
        out.println ("<HTML> <HEAD> <TITLE>");
        out.println ("PARAMETROS PASADOS </TITLE> </HEAD>");
        out.println ("<BODY>");
        out.println ("<I>Los parametros pasados son los"+
            "siguientes:</I><BR>");
        out.println ("<TT>"+parametros+"<TT>");
        out.println ("</BODY> </HTML>");
        out.close();
    } // fin del método doGet
} // fin de clase.
```

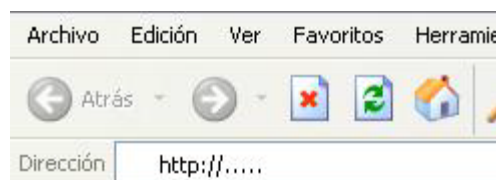
Recuperación de datos enviados en la petición

2. Cabeceras de una petición

Tipos de cabeceras

Son opcionales excepto Content-Length, que es requerida sólo para peticiones POST.

Aquí tenemos las **cabeceras** más comunes:



Leer cabeceras es fácil, sólo llamamos al método **getHeader** de **HttpServletRequest**, que devuelve un **String** si se suministró la cabecera en esta petición, y **null** si no se suministró.

Para ver todos los detalles sobre las cabeceras HTTP, puedes ver las especificaciones en <http://www.w3.org/Protocols/>

Sin embargo, hay un par de cabeceras que se usan de forma tan común que tienen **métodos** de acceso especiales.

El método **getCookies** devuelve el contenido de la cabecera Cookie, lo analiza y lo almacena en un array de objetos Cookie.

Los métodos **getAuthType** y **getRemoteUser** dividen la cabecera Authorization en sus componentes.

Los métodos **getDateHeader** y **getIntHeader** leen la cabecera específica y la convierten a valores Date e int, respectivamente.

En vez de buscar una cabecera particular, podemos usar **getHeaderNames** para obtener una Enumeration de todos los nombres de cabecera de esta petición particular.

Finalmente, además de buscar las cabeceras de petición, podemos obtener información sobre la propia línea de petición principal.

Recuperación de datos enviados en la petición

El método `getMethod` devuelve el método de petición principal (normalmente GET o POST, pero son posibles cosas como HEAD, PUT, y DELETE).

El método `getRequestURI` devuelve la URL (la parte de la URL que viene después del host y el puerto, pero antes de los datos del formulario).

El `getRequestProtocol` devuelve la tercera parte de la línea de petición que generalmente es "HTTP/1.0" o "HTTP/1.1".

Métodos para recoger las cabeceras HTTP

Enumeration `getHeaderNames()`

Devuelve una enumeración con los nombres de todas las cabeceras de la petición HTTP.

`String getHeader(String name)`

A partir del nombre de la cabecera HTTP devuelve el valor.

