

Ejercicios de introducción a los servlets

Índice

1 Servlet básico.....	2
2 Servlet que muestra la fecha y hora actuales.....	2
3 Servlet que muestra parámetros de inicio.....	3
4 (*) Configurar logging en servlets.....	3

1. Servlet básico

Para comenzar vamos a crear un proyecto web desde cero, y en él un servlet sencillo de tipo *Hola Mundo*. Seguiremos los siguientes pasos:

a) Crear un nuevo proyecto web dinámico con Eclipse, de nombre `cw-sesion01-hola`.

b) Crear un nuevo servlet dentro del proyecto, en un paquete

`es.ua.jtech.cw.sesion01.hola`, con nombre `HolaMundoServlet`. Mapearemos el servlet a la dirección `/HolaMundoServlet`.

c) Introducir en el método `doGet` del servlet el código para que muestre como salida el texto "Hola Mundo":

```
PrintWriter out = response.getWriter();
out.println ("Hola Mundo");
```

d) Ejecutar la aplicación web en Tomcat y comprobar que el servlet funciona correctamente..

2. Servlet que muestra la fecha y hora actuales

Completar el servlet `es.ua.jtech.cw.sesion01.ejercicios.FechaServlet` de la aplicación `cw-sesion01` para que, tanto por GET como por POST, muestre una página HTML con la fecha y hora actuales en una cabecera `<H3>`, y en el `<TITLE>` de la página. Para ello podéis utilizar la clase `java.util.Date`, y sacar por la salida del servlet la hora en formato cadena:

```
public void doGet(...) throws ...
{
    String fecha = "" + new java.util.Date();
    response.setContentType(...);
    out = response.getWriter();
    ... // sacar la fecha tanto en el TITLE como en una
cabecera H3
}
```

Una vez hecho, configurad el descriptor de la aplicación para que el servlet se mapee a la dirección `/fechaHora`.

Ayuda

Cuando tengamos un servlet ya implementado, pero sin configurar, podemos utilizar Eclipse para añadir automáticamente la configuración necesaria al descriptor de despliegue. Para ello entraremos en el asistente de crear nuevo servlet (*New > Servlet*), y marcamos en él la casilla *Use an existing Servlet class or JSP*. Una vez hecho esto, seleccionaremos la clase en la que se encuentra nuestro servlet y pulsamos sobre *Next*. Ahora podremos especificar la ruta a la que queremos mapear el servlet, y Eclipse añadirá la configuración necesaria al fichero `web.xml`.

3. Servlet que muestra parámetros de inicio

Crear un servlet `es.ua.jtech.cw.sesion01.ejercicios.ParamIniServlet` en la aplicación `cw-sesion01` que muestre en una tabla el nombre y el valor de todos los parámetros de inicio que se tengan configurados para ese servlet en el fichero descriptor (`web.xml`). La tabla tendrá dos columnas: una con el nombre del parámetro y otra con el valor.

Una vez hecho, probadlo añadiéndole en el fichero `web.xml` 3 parámetros de inicio con nombres `param1`, `param2` y `param3` y valores `val1`, `val2` y `val3`. Para ello deberéis dar un nombre al servlet (el nombre es arbitrario).

NOTA: para recorrer todos los parámetros de inicio, deberéis utilizar el método `init(...)` que tiene un parámetro `ServletConfig`, y utilizar dicho parámetro para obtener los nombres de los parámetros, y recorrerlos uno a uno para obtener su valor:

```
ServletConfig sc;

public void init(ServletConfig s) throws ServletException
{
    super.init(s);
    sc = s;
}

public void doGet(...) throws...
{
    java.util.Enumeration nombres = sc.getInitParameterNames();
    while (nombres.hasMoreElements())
    {
        String nombre = (String)(nombres.nextElement());
        String valor = sc.getInitParameter(nombre);
        ... // Mostrar nombre y valor en una tabla
    }
}
```

4. (*) Configurar logging en servlets

En la aplicación `cw-sesion01` tenemos dos servlets, `ServletLog4J1` y `ServletLog4J2` en el paquete `es.ua.jtech.cw.sesion01.ejercicios`. Queremos configurar las librerías de logging para poder ver los mensajes que emiten. Se pide:

- Comprobar que las librerías de logging de *commons-logging* y *log4j* están correctamente copiadas en `WebContent/WEB-INF/lib` (o en las librerías del proyecto)
- Añadir los ficheros de configuración pertinentes `commons-logging.properties` y `log4j.properties` en una carpeta de fuentes llamada `resources` para que saquen los mensajes de ambos servlets (de tipo INFO o superior) a un fichero `C:/errores.log`, con el formato:

dd/MM/aaaa hh:mm:ss - prioridad - texto del mensaje - salto de línea

- El servlet `ServletLog4J2` no saca mensajes de log. Añadid las líneas de código necesarias para que saque un mensaje de tipo INFO cuando empiece a procesar la petición (al inicio del `doGet`) y otro cuando la termine, anotando el tiempo transcurrido entre ambos mensajes (puede serte de utilidad el método `System.currentTimeMillis()` de Java).

Probad a llamar a los servlets `ServletLog4J1` o `ServletLog4J2` alguna vez, y que generen logs en el fichero `errores.log` que viene por defecto en el fichero de configuración.

