

Sistemas Web

CURSO 2021/2022

PASO 3.- Servlet para testear las clases de acceso a la base de datos



Web Sistemak by [Oskar Casquero](#) & [María Luz Álvarez](#) is licensed under a [Creative Commons Reconocimiento 4.0 Internacional License](#).

Funcionamiento del TestServlet:

- Analizará si en las solicitudes HTTP que recibe existe el parámetro **type**:
 1. Si el parámetro **type** tiene valor **registerUser**, entonces leerá los parámetros **email**, **password** y **username**; y con estos datos ejecutará **setUserInfo()**. Si el almacenamiento de datos se realiza correctamente, devolverá al cliente web la respuesta http (texto plano, no HTML) con la cadena *"El almacenamiento de datos se ha realizado correctamente"*.
 2. Si el parámetro **type** tiene valor **getUsername**, entonces se leerán los parámetros **email** y **password** y con estos datos se ejecutará **getUsername()**. Si la extracción de datos se realiza correctamente, devolverá al cliente web una respuesta http con el campo **username** (texto plano, no HTML).
 3. Si el parámetro **type** tiene valor **registerMessage**, entonces se leerán los parámetros **username** y **message** y con estos datos se hará **setMessageInfo()**. Si el almacenamiento de datos se realiza correctamente, devolverá al cliente web la respuesta http (texto plano, no HTML) con la cadena *"El almacenamiento de datos se ha realizado correctamente"*.
 4. Si el parámetro **type** tiene valor **getMessages**, entonces se hará **getAllMessages()**. Si la extracción de datos se realiza correctamente,
 - Si el parámetro **format** de la solicitud es **json**, los datos se devolverán al cliente en formato json:

```
[ {"username": "mariluz", "mezua": "¡Hola!"},  
  {"username": "oskar", "mezua": "Kaixo!"} ]
```
 - Si en la petición el parámetro **format** tiene el valor de **html**, los datos se devolverán en formato HTML mediante un documento JSP.

TestServlet

1.- Creamos un sevlet Vacio

```
package shareinfo;

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

public class TestServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public TestServlet(){
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        doGet(request, response);
    }
}
```

TestServlet

1.- Creamos un sevlet Vacio

2.- Declaramos en fichero de despliegue web.xml

```
<servlet>
  <servlet-name>TestServlet</servlet-name>
  <servlet-class>shareinfo.TestServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>TestServlet</servlet-name>
  <url-pattern>/servlet/TestServlet</url-pattern>
</servlet-mapping>
```

Añadir a web.xml

***Validar el documento XML
antes de continuar para
asegurarnos que esta bien.***

TestServlet

1.- Creamos un sevlet Vacio

2.- Declaramos en fichero de despliegue web.xml

3.- Conexión con base de datos

```
package shareinfo;
```

```
import java.io.IOException;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import helper.db.*;
```

```
public class TestServlet extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private MySQLdb mySQLdb;
```

```
    public TestServlet(){
```

```
        super();
```

```
        mySQLdb = new MySQLdb();
```

```
    }
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        doGet(request, response);
```

```
    }
```

```
}
```

doGet() en siguiente
página

1.- Si el parámetro *type* tiene valor *registerUser*, entonces leerá los parámetros *email*, *password* y *username*; y con estos datos hará *setUserInfo()*. Si el almacenamiento de datos se realiza correctamente, devolverá al cliente web la respuesta *http* (texto plano, no *HTML*) con la cadena "*El almacenamiento de datos se ha realizado correctamente*".

TestServlet

1.- Creamos un sevlet Vacio

2.- Declaramos en fichero de despliegue *web.xml*

3.- Conexión con base de datos

4.- Función *doGet()*

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("----> doPost() de TestServlet");
    PrintWriter http_out=response.getWriter(); //Escribir contenido respuesta
    String type = request.getParameter("type"); //leemos el parámetro type de la respuesta
```

```
    if (type != null ) {
        if (type.equals("registerUser")) {
            System.out.println("---- Solicitado registrar un usuario");
            String email = request.getParameter("email");
            String password = request.getParameter("password");
            String username = request.getParameter("username");
            if (email != null && password != null && username != null ) {
                System.out.println("---- Parametros: "+ email + " - " + password + " - " + username );
                mySQLdb.setUserInfo(email, password, username);
                http_out.println("El almacenamiento se ha realizado correctamente");
            }
            else { http_out.println("No se han enviado bien los parámetro");
        }
    }
    } else {
        http_out.println("No se ha enviado el parámetro type");
    }
    System.out.println("<--- doPost() de TestServlet");
}
```

TestServlet

- 1.- Creamos un servlet Vacio**
- 2.- Declaramos en fichero de despliegue web.xml**
- 3.- Conexión con base de datos**
- 4.- Función doGet()**

Antes de continuar con el resto de las opciones del Testservlet, vamos a probarlo. Arranca el servidor y en el navegador prueba:

- <http://localhost:8080/ShareInfo/servlet/TestServlet?type=registerUser&email=mi@mail.com&username=luz&password=789>
- <http://localhost:8080/ShareInfo/servlet/TestServlet?type=registerUser&email=tu@mail.com&password=789>

Estamos enviando una petición GET con los parámetros en la URI.

Con Burp prueba a realizar la misma petición con POST:

POST /ShareInfo/servlet/TestServlet HTTP/1.1

Host:localhost:8080

Content-type:application/x-www-form-urlencoded

Content-Length: 68

[type=registerUser&email=su@mail.com&username=mluz&password=777](http://localhost:8080/ShareInfo/servlet/TestServlet?type=registerUser&email=su@mail.com&username=mluz&password=777)

¿Tenemos que cambiar algo en el programa?