

Materia: LABORATORIO DE PROGRAMACIÓN PARA SERVIDORES WEB	Semestre: Sexto	Grupo: IS6A
Docente: Harim Castellanos Altamirano	Licenciatura: LICENCIATURA EN INGENIERÍA EN SOFTWARE Y SISTEMAS COMPUTACIONALES	
	Periodo: ENERO - JULIO 2017	

EJERCICIO UNO – HttpServletRequest para leer todos los parámetros

Es realizado con el uso del método **getParameterNames** de la clase `HttpServletRequest` y regresa un objeto del tipo `Enumeration` que contiene el nombre de los parámetros en un orden no especificado

El método **getParameterValues(String nombre parametro)** regresa un objeto de tipo **String[]** con los valores de los parámetros

En el proyecto Eje<HCA>P1 que ha estado trabajando durante las clases copiar las carpetas CSS y JS entregadas en su memoria USB

Crear el archivo de página web ***MostrarParametrosFormPost.html*** en raíz de EjeHCAP1. El código de la página es el siguiente:

```
<!DOCTYPE>
<HTML>
  <HEAD>
    <TITLE>Formulario usando POST</TITLE>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </HEAD>
  <BODY BGCOLOR="#FDF5E6">
    <div class="container">
      <H2 ALIGN="CENTER">Ejemplo de FORMULARIO usando POST</H2>
      <FORM ACTION="./servlet/TodosParametros" METHOD="POST" class="form-horizontal">
        <div class="form-group">
          <label class="control-label col-sm-2" for="id">Id:</label>
          <div class="col-sm-10">
            <INPUT TYPE="TEXT" NAME="id" class="form-control">
          </div>
        </div>
        <div class="form-group">
          <label class="control-label col-sm-2" for="descripcion">Descripcion:</label>
          <div class="col-sm-10">
            <INPUT TYPE="TEXT" NAME="descripcion" class="form-control">
          </div>
        </div>
      </FORM>
    </div>
  </BODY>
</HTML>
```

```
<div class="form-group">
  <label class="control-label col-sm-2" for="Precio">Precio:</label>
  <div class="col-sm-10">
    <INPUT TYPE="TEXT" NAME="precio" class="form-control">
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="Nombre">Nombre:</label>
  <div class="col-sm-10">
    <INPUT TYPE="TEXT" NAME="nombre" class="form-control">
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="ApePaterno">Ape. Paterno:</label>
  <div class="col-sm-10">
    <INPUT TYPE="TEXT" NAME="ApePaterno" class="form-control">
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="ApeMaterno">Ape. Materno:</label>
  <div class="col-sm-10">
    <INPUT TYPE="TEXT" NAME="ApeMaterno" class="form-control">
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="Direccion">Dirección:</label>
  <div class="col-sm-10">
    <TEXTAREA NAME="direccion" ROWS=3 COLS=40 class="form-control"></TEXTAREA>
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="tdc">Tarjeta de Crédito:</label>
  <div class="col-sm-10">
    <div class="radio">
      <label><INPUT TYPE="RADIO" NAME="tdc" VALUE="Visa">Visa</label>
    </div>
    <div class="radio">
      <label><INPUT TYPE="RADIO" NAME="tdc" VALUE="MasterCard">MasterCard</label>
    </div>
    <div class="radio">
      <label><INPUT TYPE="RADIO" NAME="tdc" VALUE="Amex">American Express</label>
    </div>
    <div class="radio">
      <label><INPUT TYPE="RADIO" NAME="tdc" VALUE="Prosa">Prosa</label>
    </div>
    <div class="radio">
      <label><INPUT TYPE="RADIO" NAME="tdc" VALUE="Java SmartCard">Java SmartCard</label>
    </div>
  </div>
</div>
</div>
```

```
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="numTdc">Número de tarjeta de crédito:</label>
        <div class="col-sm-10">
            <input type="password" name="numTdc" class="form-control">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="numTdc">Repetir Número de tarjeta de crédito:</label>
        <div class="col-sm-10">
            <input type="password" name="numTdc" class="form-control">
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <button type="submit" class="btn btn-default">Enviar</button>
        </div>
    </div>
</FORM>
</div>
</BODY>
</HTML>
```

El resultado del código anterior debe verse como sigue:

Ejemplo de FORMULARIO usando POST

Id:	<input type="text"/>
Descripcion:	<input type="text"/>
Precio:	<input type="text"/>
Nombre:	<input type="text"/>
Ape. Paterno:	<input type="text"/>
Ape. Materno:	<input type="text"/>
Dirección:	<input type="text"/>
Tarjeta de Crédito:	<div><input type="radio"/> Visa <input type="radio"/> MasterCard <input type="radio"/> American Express <input type="radio"/> Prosa <input type="radio"/> Java SmartCard</div>
Número de tarjeta de crédito:	<input type="text"/>
Repetir Número de tarjeta de crédito:	<input type="text"/>
	<input type="button" value="Enviar"/>

Crear un servlet llamado **TodosParametros.java** dentro del paquete *psw* del proyecto EjeHCAP1. En el código siguiente debe poner especial atención en los métodos que están siendo utilizados. El código únicamente muestra el método **doGet** usted debe hasta este punto saber todo lo que involucra crear un Servlet y la configuración mínima para ponerlo en marcha.

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String docType = "<!DOCTYPE>\n";
    String title = "Leer todos los parametros en Request";
    out.println(docType + "<HTML>\n" +
        "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
        "<BODY BGCOLOR=\"#FDF5E6\">\n" +
        "<H1 ALIGN=CENTER>" + title + "</H1>\n" +
        "<TABLE BORDER=1 ALIGN=CENTER>\n" +
        "<TR BGCOLOR=\"#FFAD00\">\n" +
        "<TH>Nombre del parametro<TH>Valor de los parametros");
    Enumeration nombresParametros = request.getParameterNames();
    while(nombresParametros.hasMoreElements()) {
        String nombreDelParametro = (String)nombresParametros.nextElement();
        out.print("<TR><TD>" + nombreDelParametro + "\n<TD>");
        String[] valoresParametros = request.getParameterValues(nombreDelParametro);
        if (valoresParametros.length == 1) {
            String paramValue = valoresParametros[0];
            if (paramValue.length() == 0)
                out.println("<I>No hay valor</I>");
            else
                out.println(paramValue);
        } else {
            out.println("<UL>");
            for(int i=0; i<valoresParametros.length; i++) {
                out.println("<LI>" + valoresParametros[i]);
            }
            out.println("</UL>");
        }
    }
    out.println("</TABLE>\n</BODY></HTML>");
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    doGet(request, response);
}
```

Modificar el archivo web.xml como corresponde
Compilar las clases java y mostrar el ejemplo al maestro

EJERCICIO DOS

Utilizando la página web de su tarea **login.html** crear un Servlet llamado **LoginControlador.java** que haga la lectura de los parámetros **usuario** y **contraseña**.

Los parámetros leídos deben ser comparados con cadenas en código duro que usted defina como usuario y contraseña correcto, es decir, si los datos ingresados son correctos crear un mensaje “Bienvenido usuario: <valor del parámetro usuario>”, por el contrario si los datos son incorrectos crear un mensaje “Usuario incorrecto: <valor del parámetro usuario>”.

Ya sea que el usuario sea correcto o no debe crear la página web de salida creada en el método *doGet* como se ha venido haciendo en clase para mostrar los mensajes.

EJERCICIO TRES

Utilizando la página web de su tarea **altaUsuario.html** crear un Servlet llamado **UsuarioControlador.java** que haga la lectura de los parámetros correspondientes.

El Servlet debe mostrar una tabla con todos los parámetros recibidos. La tabla de dos columnas la primera es el nombre del parámetro y la segunda el valor del parámetro.

EJERCICIO CUATRO

Eclipse IDE for Java EE Developers

1. En el link <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplerr>
2. Seleccionar la arquitectura del sistema operativo y descargar el software



eclipse

GETTING STARTED MEMBERS PROJECTS MORE ▾

DOWNLOAD

HOME / DOWNLOADS / PACKAGES / ECLIPSE IDE FOR JAVA EE DEVELOPERS

Eclipse IDE for Java EE Developers

Package Description

Tools for Java developers creating Java EE and Web applications, including a Java IDE for Java EE, JPA, JSF, Mylyn, EGit and others.

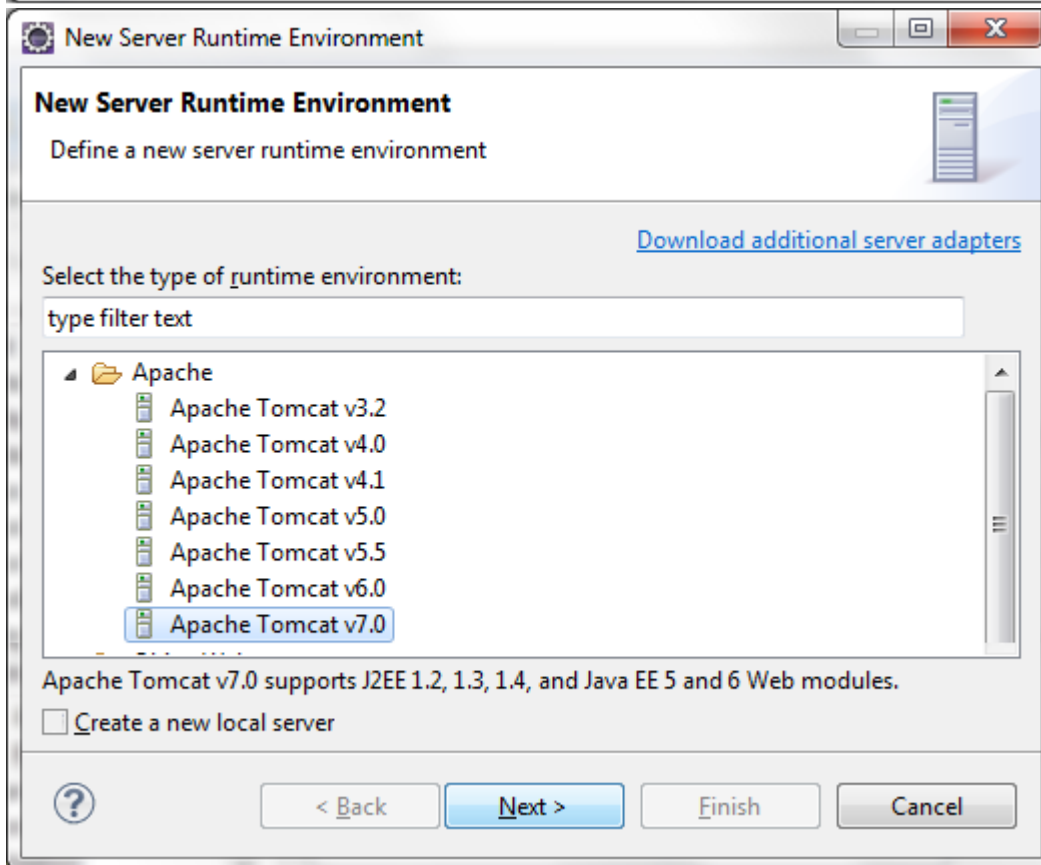
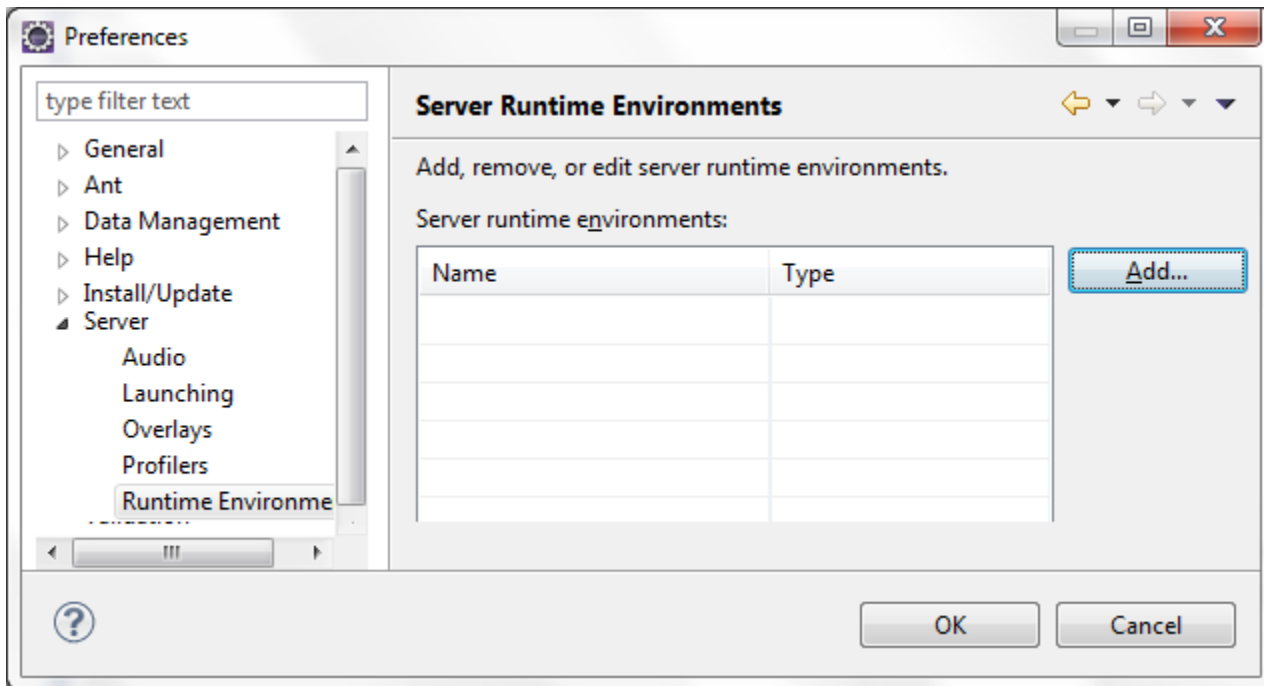
This package includes:

- Data Tools Platform

Download Links

- Windows 32-bit
- Windows 64-bit
- Mac OS X (Cocoa) 32-bit
- Mac OS X (Cocoa) 64-bit
- Linux 32-bit
- Linux 64-bit

3. Descomprimir software en ruta de trabajo. Doble clic en archivo *eclipse.exe* dentro del directorio
 4. Seleccionar Windows->Preferences->Server->RunTimeEnvironment->Add->Seleccionar Tomcat 7.0
- En caso de no tener Tomcat 7. Descargarlo de la página del Apache o pedirlo al maestro mediante su USB



Indicar la ruta de la carpeta raíz de Tomcat 7. Clic en finalizar

New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name:
Apache Tomcat v7.0

Tomcat installation directory:
C:\Informacion\desarrollo\apache-tomcat-7.0.2 Browse...

apache-tomcat-7.0.12 Download and Install...

JRE:
Workbench default JRE Installed JREs...

Preferences

type filter text

- General
- Ant
- Data Management
- Help
- Install/Update
- Java
- Server
 - Audio
 - Launching
 - Overlays
 - Profilers
 - Runtime Environments

Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type	
Apache Tomcat v7.0	Apache Tomcat v7.0	

Add...
Edit...

? OK Cancel