

Uso de JavaBean_II

Introducción

- Vamos a ver una nueva utilidad de los JavaBean: aplicaciones web siguiendo el MVC
- ¿Qué necesitamos saber?
 - JSP
 - Servlets
 - MVC
- Todo lo iremos viendo poco a poco.
- Primero → preparar nuestro entorno para trabajar con este tipo de aplicaciones.

Preparando Eclipse

- Help → Available Software → Del desplegable optamos por “--All available Sites --”
- Seleccionamos:



- Tardará un buen rato en calcular dependencias... → Aceptamos licencias y “Finish”
- Habrá que reiniciar Eclipse.

Instalando Tomcat

- Acudimos a <http://tomcat.apache.org/>
- Nos descargamos la versión “Core” 8.5.40
- Activamos la perspectiva “Servers”.
-

Primeros pasos

- Creamos proyecto del tipo “Dynamic Web Project”. Llamaremos al proyecto “WebApp”. Pediremos que Eclipse genere el “descriptor” del proyecto:
web.xml
- Añadimos una página html en **New** → **HTML**.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><html><head><META  
http-equiv="Content-Type" content="text/html; charset=utf-8"></head>  
<body>  
<h1 align="center">Gestión</h1>  
<p align="center">  
    <a href="http:///WebApp/Controlador?accion=listado" target="_blank"> Listado </a>  
</p>  
</body></html>
```

- Ejecuta y observa el resultado.

Añadimos Servlet

- Creamos un nuevo paquete en “Source” que llamaremos “control”
- Añadimos **New** → **Servlet**. Reescribimos el método **doGet**:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    String op = request.getParameter("accion");  
    if (op.equals("alta")) {  
        System.out.println("depurando");  
        response.sendRedirect("alta.jsp");  
    }  
}
```

- Estamos reenviando a “alta.jsp”

Web.xml

- Añadimos las siguientes líneas al descriptor del proyecto:

```
<servlet>
  <servlet-name>Controlador</servlet-name>
  <servlet-class>Controlador</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Controlador</servlet>
  <url-pattern>/Controlador</servlet>
```

JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%><!DOCTYPE html>
<html><head><meta charset="UTF-8">
<title>VayaTela</title>
</head>
<jsp:useBean id="depart" scope="request" class="miWeb.Departamentos"></jsp:useBean>
<jsp:setProperty property="*" name="depart"/>
<%
if(request.getParameter("deptno")!=null){%>
<jsp:forward page="/Controlador?accion=insertar"></jsp:forward>
<% } %>
<body>
<center><h2>Entrada </h2>
<form method="post">
<p> Num departamento:
<input name="deptno" required type="number" min="1" max="99" /> </p>
<p> Nombre:
<input name="dnombre" required type="text" size="15" maxlength="15" /> </p>
<p> localidad:
<input name="loc" required type="text" size="15" maxlength="15"/> </p>
<input type="submit" name="insertar" value="insertar" />
</form>
<a href="index.html"> Inicio</a>
</center>
</body>
</html>
```


Modelo

- Falta añadir nuestro modelo de datos. En concreto, necesitamos un Bean para representar a la clase “Departamento”.

```
public class Departamentos {  
    private int deptno;  
    private String dnombre;  
    private String loc;  
  
    String modificar;  
    String eliminar;  
    String insertar;  
    .... Constructores y getters/setters ....  
}
```

Acceso a datos

- Utilizaremos la base de datos NeoDatis. Hay que añadir los drivers a las rutas del proyecto. Debe poder acceder también el servidor Tomcat
- Falta añadir en el controlador el acceso a la bbdd para guardar datos. Habrá que modificar el doGet del Controlador. La modificación empezará de la siguiente manera:

```
if (op.equals("insertar")) {  
    control.Departamentos dep = (control.Departamentos) request.getAttribute("depart");  
    .....
```