Proyecto de Diseño y Construcción de Componentes.

JavaServer Faces (JSF)

Profesor: Mario Agüero

Realizado por:

|  |
| --- |
| Yuliana Arias |
| Chris Esquivel |
| Jose Pablo Carballo |

**2014**

4/15/2014

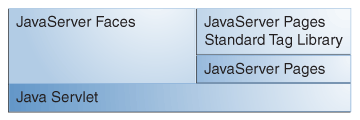
JavaServer Faces (JSF)

JavaServer Faces es un framework de interfaz de componentes de usuarios del lado del servidor para las aplicaciones web basadas en la tecnología en Java, es una especificación desarrollada por la Java Community Process. Los principales componentes de la tecnología JavaServer Faces son:

* APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada.
* Una librería de etiquetas JavaServer Pages (JSP) personalizadas que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.

Este modelo de programación facilita la construcción y mantenimiento de aplicaciones Web con UIs del lado del servidor; se pueden conectar eventos generados en el cliente a código de la aplicación en el lado del servidor.

Una de las grandes ventajas de la tecnología JavaServer Faces es que ofrece una clara separación entre el comportamiento y la presentación. Contiene una arquitectura donde se puede manejar el estado de los componentes, procesamiento de los datos y el manejo de los eventos. Como parte de la tecnología JavaServer Faces se encuentra la tecnología Facelets.



***Java Web Application Technologies***

Facelets

Facelets es un framework de declaración de página, poderoso pero ligero, que es usado para construir vistas de JavaServer Faces usando plantillas de estilo de HTML y construyendo árboles de componentes.

Las características de Facelets incluyen son algunas de las siguientes:

* Uso de XHTML para crear páginas web.
* Soporte para librerías de etiquetas Facelets que se suman a las librerías de JavaServer Faces y JSTL.
* Soporte para el Lenguaje de Expresiones EL.
* Plantillas para componentes y páginas.

Las ventajas que Facelets incluye para el desarrollo de proyectos de gran escala son las siguientes:

* Soporte para reutilización de código por medio de plantillas y componentes compuestos.
* Extensibilidad funcional de componentes y otros objetos del lado del server por configuración.
* Tiempo de compilación rápido.
* Validación de EL en tiempo de compilación.

Relación entre JFS y Glassfish

Glassfish Server es un servidor open-source utilizado para crear y entregar aplicaciones basadas en tecnologia Java. En sí, Glassfish es una implementación de la especificación Java EE, considerada la implementación "referencia" pues implementa la totalidad de especificaciones de Java EE y Java EE Web Profile. Por ende, el framework JavaServer Faces es parte de las especificaciones que implementa Glassfish. El servidor se integra con IDE's como Netbeans y Eclipse, a su vez que ofrece una gama de herramientas que habilitan un modelo eficiente para desarrollar y publicar aplicaciones.

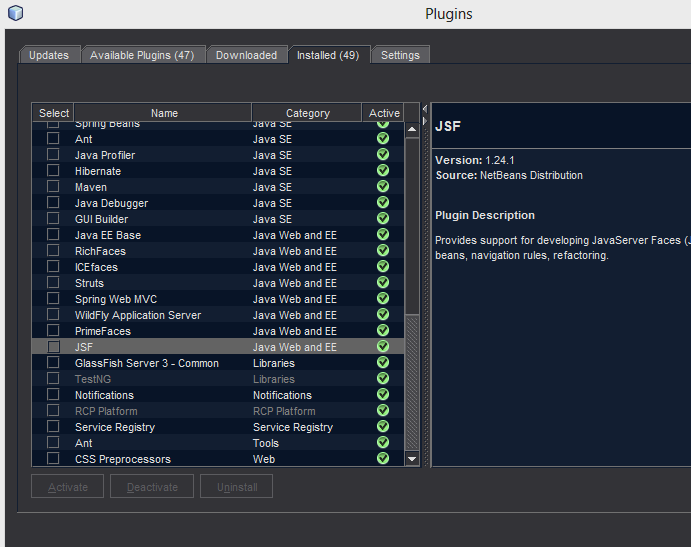
Guía de configuración de JSF 2.x usando Glassfish en Netbeans 7.x, 8.x IDE

Requerimientos:

* NetBeans 7.x o superior
* Glassfish Server 3 NetBeans Plugin v1.58.1
* JSF NetBeans Plugin v1.58.1
* JDK 6 o superior

Paso 1

Verificar que los plugins en NetBeans estén activos y correctamente instalados.



Si no; Ir a “Available Plugins” tab en la ventana de Plugins. (Plugins se localiza en Tools>Plugins) e instalar los 2.

Paso 2

Como ejemplo, vamos a usar este proyecto:

<https://netbeans.org/projects/samples/downloads/download/Samples%252FJavaEE%252FjsfDemo.zip>

Descomprimir y abrir el proyecto en NetBeans.

Paso 3

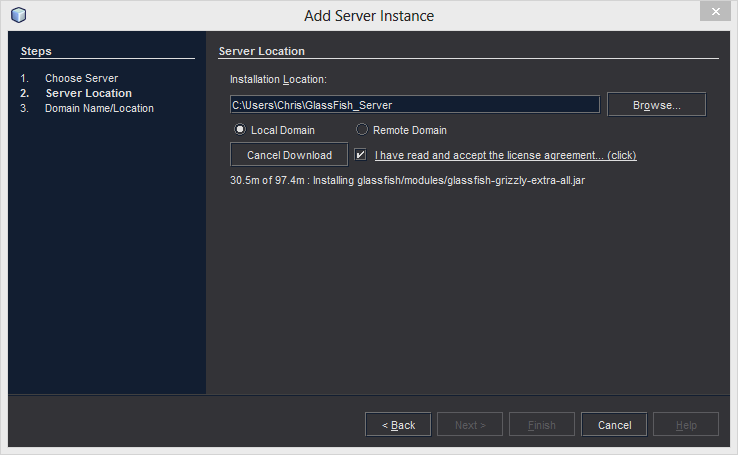
Si se corre el proyecto va a dar un error, ya que no hay un Java EE Server configurado para el proyecto; para resolver esto se le da click derecho a el proyecto y se selecciona “Resolve Missing Server Problem”.

Si no, saltar al paso 6.

Paso 4

Seleccionar “Add Server” en el pop­up y seleccionar Glassfish Server.

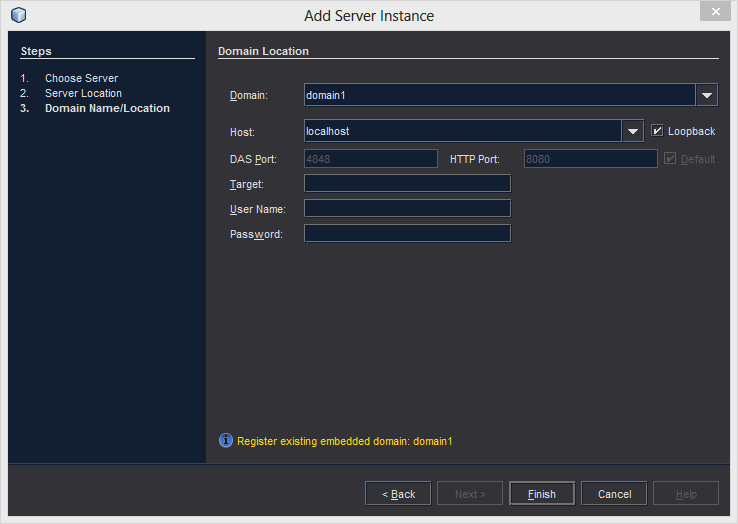
Si es la primera vez que se configura este servidor; se debe especificar la ruta local del mismo. En este caso “C:\Users\Chris\GlassFish\_Server”. Seguidamente hacerle click al checkbox y descargar  Glassfish Server 4.0 (Aprox. 100MB).



Una vez que se descarga, hacer click en Next.

Paso 5

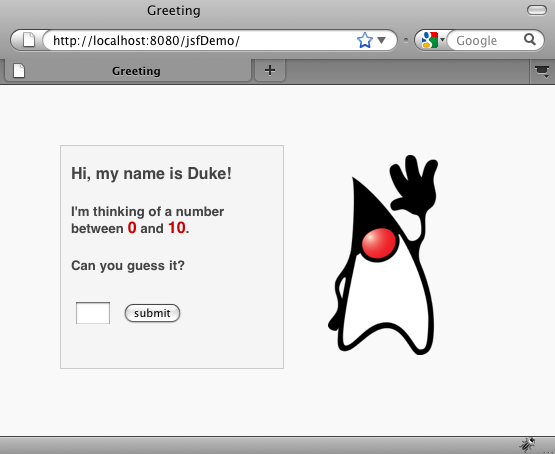
Dejar la configuracion default en la pantalla de Domain Location. Hacer click en Finish.



Paso 6

Si se configuró el servidor por primera vez, el primer Run el proyecto puede durar hasta 5 minutos en correr. (esto es normal).

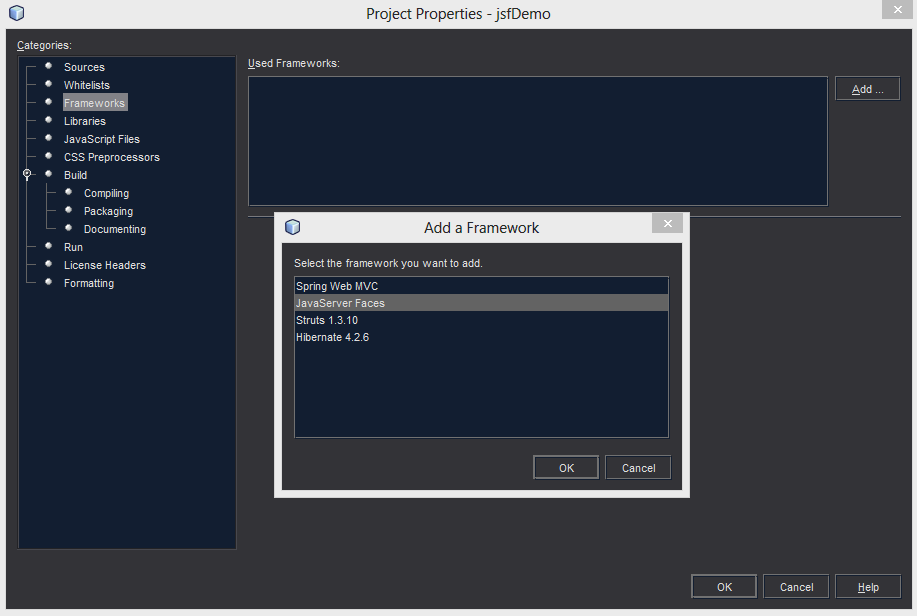
Seguidamente en el browser de default de la computadora debería abrirse el index.html del proyecto.



Al hacer click en Submit; debería salir un mensaje que dice “[response here]”.

Paso 7

Seleccionar las propiedades del proyecto y agregar JSF al proyecto en la sección de Frameworks.



Dar click en OK.

Paso 8

En el tab de Libraries seleccionar el Server Library de JSF 2.x.

En el tab de configuración seleccionar que el JSF URL Pattern sea “/faces/\*”

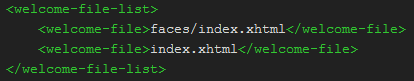
En el tab de configuración seleccionar que el Preferred Page Language sea Facelets.

Dar click en OK.

Paso 9

El web.xml del proyecto se actualizo sin darnos cuenta…

Hay que asegurarse de que el tag de XML <welcome­file> coincida con el URL Pattern de JSF.



Solo debe existir 1 <welcome­file>, en este caso se debe borrar la línea 3. (ya que es la que tiene un JSF  URL Pattern quivocado)

Guardar cambios en el XML y ya el proyecto esta listo para desarrollar en bajo el framework de JSF.

**Nota:**

Para correr una aplicación con JFS, es necesario que la máquina cumpla con estos requisitos:

**Servidor:**

* Intel Core 2 Duo o equivalente AMD CPU
* 2 GB de RAM o más.
* Suficiente espacio para un sistema operativo, servidor de la aplicación, la aplicación, los log files y

otros.

* Windows XP o superior, Mac OS 10.8 o superior o alguna distribución de Linux.
* Conexión de red por la cual las páginas web serán servidas, y una dirección IP para que el cliente

pueda conectarse.

* Java Development Kit, o OpenJDK.
* Java EE server, dependiendo de la aplicación, JBoss o Tomcat o Glassfish o Jetty o Jonas.

**Cliente:**

* Dominio o IP para conectarse a la aplicación.
* Un browser actualizado (Chrome, FF, IE10...)

Comparación JSF frente a Struts

JSF es uno de los posibles frameworks que se pueden utilizar para crear aplicaciones web Java. Uno de los más extendidos es Struts, un framework bien maduro, que sin embargo no forma parte del estándar J2EE.

En esencia, Struts es un controlador MVC de tipo 2, en el que la acción del formulario se liga a una clase acción, que sigue el patrón “comando” de diseño (una clase que representa un comando). El controlador de Struts sigue a su vez el patrón de “controlador frontal”, dirigiendo cada petición solicitada a la clase acción que la procesa, y pasándole los datos de la petición encapsulados en un objeto (patrón “objeto valor”).

JSF se desarrolló teniendo en mente Struts, ya que los impulsores de este desarrollo, en los que se encontraban gente como IBM y el propio creador de Struts, querían crear un nuevo framework más potente.

Prácticamente todo lo que se puede hacer con Struts tiene un equivalente JSF (hay como siempre que adaptarse al modo de trabajo de JSF, no intentar “clonar” el comportamiento de Struts en JSF), pero además utilizando JSF obtendremos una serie de ventajas:

* La arquitectura del controlador es más avanzada y flexible. Podremos hacer algunas tareas avanzadas de manera sencilla (por ejemplo, utilizando “phase listeners”).
* JSF permite definir la navegación no solo a través de los métodos de navegación de los beans, sino incluso en la propia página (navegación definida en los componentes de la página).
* JSF permite recoger los parámetros del formulario de manera más sencilla que Struts, e incorpora un lenguaje de expresiones que lo hace más simple.
* JSF soporta la creación de manejadores de eventos asociados a los componentes de la página, lo que dota a dichos componentes de gran potencia. Un ejemplo: creación de combos enlazados, en los que la elección de un elemento en el primer combo obliga a recalcular los elementos disponibles en el segundo combo, por ejemplo, en combos de países y estados o ciudades.
* JSF está pensado para la creación de interfaces de usuario avanzadas. Basta ver los nuevos frameworks extendidos JSF, como Apache MyFaces Trinidad y el resto de frameworks JSF con soporte AJAX.
* JSF tiene capacidad para ser visto en dispositivos móviles mientras que Struts está limitado a HTML y HTTP
* JSF es más reutilizable y orientado a componentes mientras que en Struts no.
* Es más fácil usar el archivo de configuración faces-config.xml que struts-config.xml ya que en JSF lo podemos interpretar de manera gráfica con algún IDE, y hacerlo más intuitivo y claro.