
Java Server Faces

Ingeniería de Software
Laboratorio

— L. en C.C. Miguel Angel Piña
Avelino —

Introducción

Introducción

JSF es un framework MVC (Modelo-Vista-Controlador) basado en el API de Servlets que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets. Antes de la especificación actual se utilizaba JSP para componer las páginas JSF.



Introducción

Entrando un poco más en detalle, JSF proporciona las siguientes características destacables:

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Conexión de los componentes gráficos con los datos de la aplicación mediante los denominados beans gestionados.
- Conversión de datos y validación automática de la entrada del usuario.

Introducción

Entrando un poco más en detalle, JSF proporciona las siguientes características destacables:

- Navegación entre vistas.
- Internacionalización.
- A partir de la especificación 2.0 un modelo estándar de comunicación Ajax entre la vista y el servidor.

Introducción

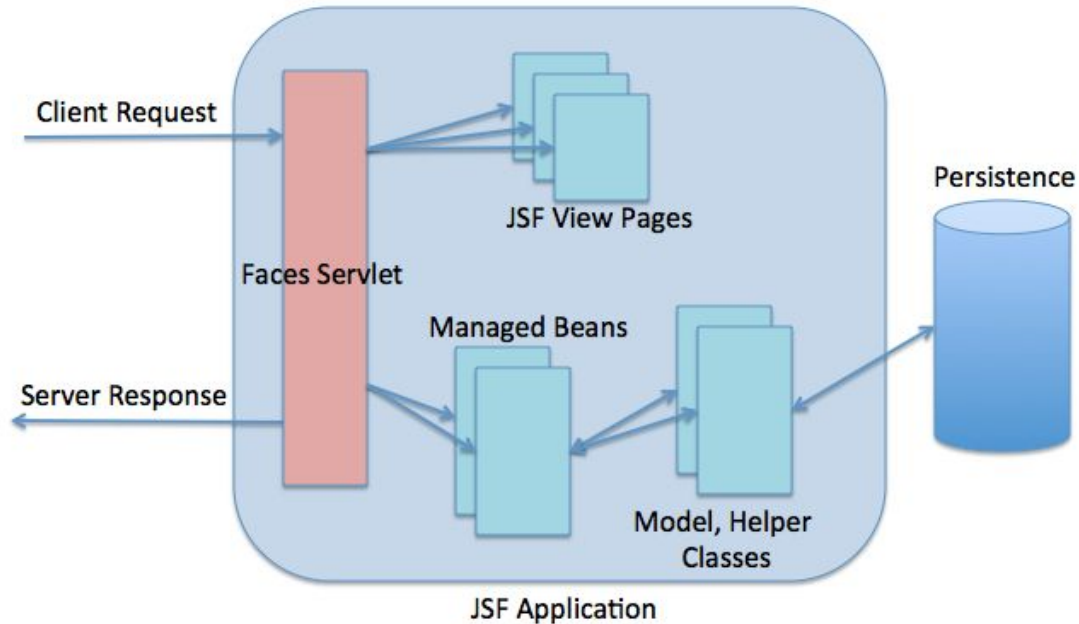
Otras características que tiene JSF son:

- Soporte para Ajax
- Componentes múltiples
- Integración con Facelets
- Gestión de recursos (hojas de estilo, imágenes, etc.)
- Facilidad de desarrollo y despliegue

¿Cómo se ejecuta?

Tal y como hemos comentado, JSF se ejecuta sobre la tecnología de Servlets y no requiere ningún servicio adicional, por lo que para ejecutar aplicaciones JSF sólo necesitamos un contenedor de servlets tipo Tomcat o Jetty.

¿Cómo se ejecuta?



Implementaciones

JSF es una especificación y, como tal, existen distintas implementaciones. Sun siempre proporciona una implementación de referencia de las tecnologías Java, que incluye en el servidor de aplicaciones GlassFish. En el caso de JSF, la implementación de referencia s las dos implementaciones más usadas son:

- Mojarra
- MyFaces

<https://docs.oracle.com/javaee/7/tutorial/jsf-intro.htm>

Extensiones

Además de las implementaciones que existen para JSF, existen extensiones al framework que provee más componentes, ya sean extensiones basadas en bibliotecas como ***jQuery*** hasta componentes más sofisticados como ***Primefaces***.

- ***Primefaces***: PrimeFaces es una biblioteca muy simple que no necesita dependencias y configuraciones.
- ***IceFaces***: Contiene diversos componentes para interfaces de usuarios más enriquecidas.
- ***RichFaces***: Agrega componentes visuales y soporte para AJAX

Extensiones

Además de las implementaciones que existen para JSF, existen extensiones al framework que provee más componentes, ya sean extensiones basadas en bibliotecas como ***jQuery*** hasta componentes más sofisticados como ***Primefaces***.

- ***jquery4jsf***: Contiene diversos componentes sobre jQuery.
- ***OpenFaces***: Biblioteca open source que contiene diferentes componentes JSF, componentes Ajax y validación por parte del cliente.

Primefaces

PrimeFaces es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces.

<http://primefaces.org/>

Integrando JSF

Integrando JSF en Maven

Para ejemplificar la integración de JSF y la facilidad con la que se puede operar con JSF, vamos a implementar un formulario que permita ***simular el registro de un usuario***.

Lo primero que hay que hacer es tomar el ejemplo de Maven que se construyó en sesiones pasadas y modificar el archivo ***pom.xml*** en la sección de dependencias agregando el siguiente código para integrar las dependencias de JSF a nuestro proyecto:

Gist: <https://goo.gl/1FKL9u>

Integrando JSF con Maven

El siguiente archivo a modificar es **web.xml**, en el siguiente Gist se encuentra el código que debe contener.

<https://goo.gl/52sQGm>

Ahora que ya tenemos el soporte de las bibliotecas de JSF dentro del proyecto, vamos a comenzar a crear la página para validar la creación de inicio de sesión. Para esto comenzaremos con el concepto de Managed Bean.

¿Qué es un Managed Bean?

Un Managed Bean es una clase que sigue la nomenclatura de los Java Beans. El objetivo de estos objetos es controlar el estado de la página web. JSF va a administrar automáticamente los managed beans.

Ahora vamos a integrar un bean que represente un usuario del sistema. Este va a estar en un archivo en la siguiente ruta:

```
src/main/java/com/miguel/proyecto/web/Usuario.java
```

El contenido está en el siguiente Gist: <https://goo.gl/C9PykT>

Integrando JSF con Maven

Después de agregar una clase que represente el modelo de nuestra aplicación, agregaremos una clase que represente el controlador. Para esto usaremos un bean manejado de JSF. Este bean lo vamos a agregar en la ruta

```
src/main/java/com/miguel/proyecto/web/RegisterController.java
```

El contenido está en el siguiente Gist: <https://goo.gl/4Bhf3b>

Integrando JSF con Maven

El último paso es agregar una página que permita probar el funcionamiento de los beans. Esta página estará en xhtml y su ruta es:

```
src/main/webapp/registro.xhtml
```

El contenido de esta página está en el Gist: <https://goo.gl/gmzrUa>

Para probar su funcionamiento, ejecutamos en una terminal:

```
mvn tomcat7:run
```

Y en el navegador:

```
http://localhost:8080/mi-primer-aplicacion-web/registro.xhtml
```