

Desarrollando Aplicaciones Web con JSF

Módulo 3 - Clase #01
Carrera Java Programmer SE 8

Introducción a Java Server Faces (JSF)

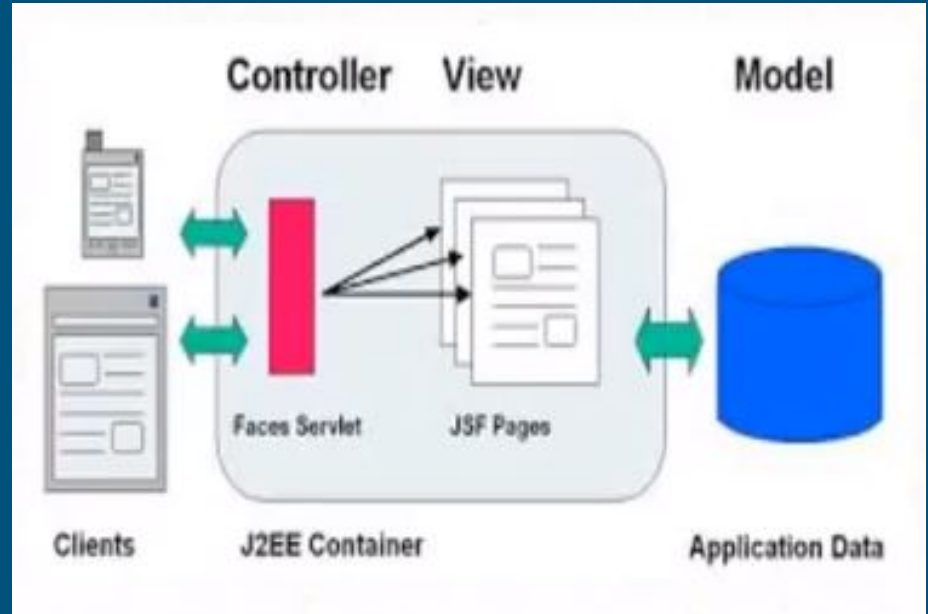
Características:

- Framework para crear aplicaciones web estándar para JEE.
- Cuenta con soporte completo en Netbeans, Eclipse, entre otros.
- Aplica el patrón de diseño MVC.
- RAD: Desarrollo de aplicaciones web ágiles.
- Render-Kids: Los componentes pueden usarse en dispositivos móviles y otros dispositivos.
- Es extensible: JSF es altamente extensible debido a su arquitectura.

JSF v 2.*

Nuevas características:

- Manejo de condiciones más inteligentes.
- Manejo de anotaciones de configuraciones.
- Soporte nativo para AJAX.
- Soporte para Facelets.
- Más componentes y validadores.



MVC en JSF



Vistas en JSF

Vistas:

- Asocia a cada vista un conjunto de objetos Java, manejados por el controlador (Managed Beans).
- Facilita la obtención, manipulación y visualización de los valores.
- Se utiliza un archivo de configuración para el controlador en formato XML.
- Es extensible.

Lenguaje de expresiones EL:

- Permite acceder a las propiedades de los BackEnd Beans.
- Se utiliza la siguiente estructura:
 - `{ Expresión a utilizar }`
 - `$ { Expresión a utilizar }`
- Por ejemplo:
 - `<h:outputText value="#{backbean.propiedad}" />`
 - `<h:inputText value="#{backbean.propiedad}" />`

Algunas Expresiones EL

| Expresión EL | Tipo de Propiedad | Valor |
|---|---|--|
| <code>bean.stringProperty</code> | String | El valor del String. |
| <code>bean.myBoolean</code> | boolean | true o false o su cadena “true” o “false”. |
| <code>bean.property.property2</code> | property: Class property2: String | El valor de la propiedad 2 del objeto property del bean. |
| <code>bean.myHashMap['madrid']</code> | Bean: Class | Devuelve el elemento de clave 'madrid' en el hashmap myhashmap del bean. |
| <code>bean.myHashMap['madrid'].property3</code> | Bean: Class myHashMap: HashMap (Map) | Devuelve la propiedad property3 del elemento de clave 'madrid' en el hashmap myhashmap del bean. |
| <code>bean.myHashMap.madrid</code> | Bean: class myHashMap: HashMap (Map) | Devuelve el elemento de clave 'madrid' en el hashmap myhashmap del bean (otra forma de acceder). |

Las etiquetas JSF

| Etiqueta | Descripción |
|-----------------|---|
| h:commandButton | Un botón al que podemos asociar una acción. |
| h:commandLink | Un enlace hipertexto al que podemos asociar una acción. |
| h:dataTable | Crea una tabla de datos dinámica con los elementos de una propiedad de tipo Array o Map del bean. |
| h:form | Define el formulario JSF en la página JSP. |
| h:inputText | Incluye un campo de texto normal. |

<https://docs.oracle.com/javaee/6/javaxserverfaces/2.0/docs/pdldocs/facelets/index.html>

Ventajas y Desventajas de Usar JSF

Ventajas:

- El código es muy parecido al HTML estándar.
- JSF se encarga de la obtención y generación de los valores de los elementos de la página.
- JSF resuelve validaciones, conversiones, mensajes de error e internacionalización (i18n).
- JSF permite introducir de JavaScript.
- JSF es extensible.

Desventajas:

- Utilizar el alicate para clavar un clavo.
- Abuso del JavaScript.
- La maquetación compleja (DataTables).

Managed Beans

General:

- Es una clase Java que sigue la nomenclatura de JavaBeans.
- Los Managed Beans no están obligados a extender de ninguna clase.

Tipos de Beans:

- Modelo: Representan el modelo
- Control: Representan el controlador
- Soporte o Helpers: Contienen código, por ejemplo convertidores
- Utilerias: Tareas genéricas como obtener el objeto HttpRequest.

Uso de los Managed Beans

Formas:

- Con anotaciones, antes del nombre de la clase:
 - `@ManagedBean`
- Cómo bean CDI (Content and Dependency Injection), antes del nombre de la clase:
 - `@Named`
 - Agregar un archivo llamado `beans.xml` en la carpeta `web/WEB-INF`
- En el archivo `faces.config.xml`
 - `<managed-bean>...</managed-bean>`

Alcance de los Managed Beans

| Scope | Descripción |
|-------------|--|
| Application | Persiste durante toda la aplicación. |
| Session | Persiste durante el tiempo de la sesión del usuario. |
| View (2.0) | persiste si es la misma vista. Útil para usar AJAX. |
| Request: | Persiste solo durante la petición del usuario. |

Ejemplo

