

Java Server Faces

Java EE 7



¿Qué es?

Beneficios

Manager Bean

Scopes



¿Qué es?

La funcionalidad proporcionada por una aplicación JavaServer Faces es similar a la de cualquier otra aplicación web Java. Una aplicación típica de JavaServer Faces incluye las siguientes partes.

Un conjunto de páginas web en las que se disponen los componentes.

Un conjunto de etiquetas para agregar componentes a la página web.

Un conjunto de beans gestionados, que son objetos ligeros, gestionados por contenedor (POJOs). En una aplicación JavaServer Faces, los beans administrados sirven como beans de respaldo, que definen propiedades y funciones para los componentes de la interfaz de usuario en una página.

Un descriptor de despliegue web (archivo web.xml).

Opcionalmente, uno o más archivos de recursos de configuración de la aplicación, como un archivo faces-config.xml, que se pueden usar para definir las reglas de navegación de la página y configurar beans y otros objetos personalizados, como componentes personalizados.

Opcionalmente, un conjunto de objetos personalizados, que pueden incluir componentes personalizados, validadores, convertidores u oyentes, creados por el desarrollador de la aplicación.

Opcionalmente, un conjunto de etiquetas personalizadas para representar objetos personalizados en la página.



¿Qué es?

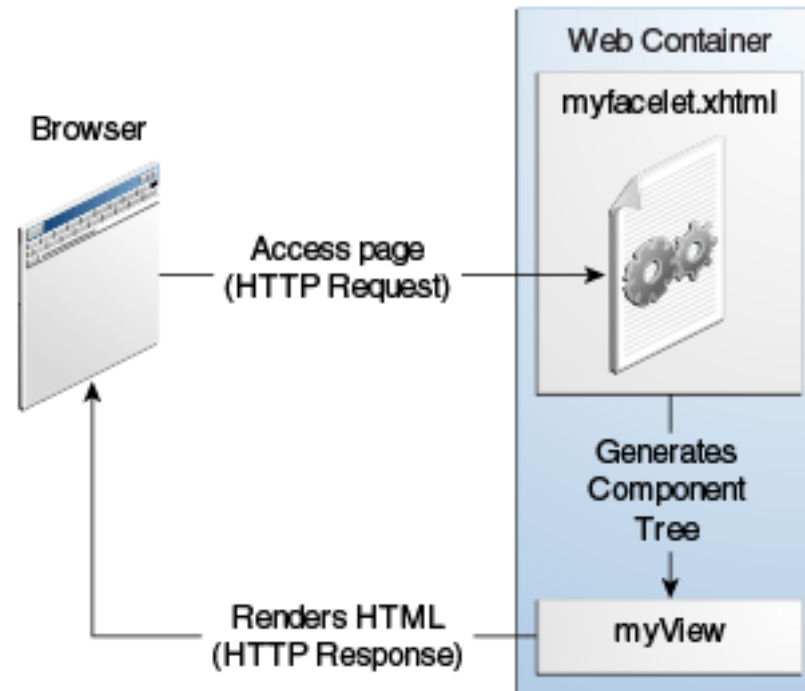
Beneficios

Manager Bean

Scopes



¿Qué es?



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

¿Qué es?

Beneficios

Manager Bean

Scopes



¿Qué es?

La página web, myfacelet.xhtml, se construye utilizando etiquetas de componente JavaServer Faces. Las etiquetas de componentes se utilizan para agregar componentes a la vista (representada por myView en el diagrama), que es la representación del lado del servidor de la página. Además de los componentes, la página web también puede hacer referencia a objetos, como los siguientes:

- Cualquier detector de eventos, validadores y convertidores registrados en los componentes.
- Los componentes de JavaBeans que capturan los datos y procesan la funcionalidad específica de la aplicación de los componentes.

A petición del cliente, la vista se representa como una respuesta. La representación es el proceso mediante el cual, según la vista del lado del servidor, el contenedor web genera un resultado, como HTML o XHTML, que puede leer el cliente, como un navegador.

¿Qué es?

Beneficios

Manager Bean

Scopes



Beneficios

Una de las mayores ventajas de la tecnología JavaServer Faces es que ofrece una separación clara entre el comportamiento y la presentación para aplicaciones web. Una aplicación JavaServer Faces puede asignar solicitudes HTTP a un manejo de eventos específico del componente y administrar componentes como objetos con estado en el servidor. La tecnología JavaServer Faces le permite crear aplicaciones web que implementan la separación más precisa del comportamiento y la presentación que tradicionalmente ofrecen las arquitecturas de UI del lado del cliente.

La separación de la lógica de la presentación también permite que cada miembro de un equipo de desarrollo de aplicaciones web se centre en una sola parte del proceso de desarrollo y proporciona un modelo de programación simple para vincular las piezas. Por ejemplo, los autores de páginas sin experiencia en programación pueden usar etiquetas de tecnología JavaServer Faces en una página web para enlazar con objetos del lado del servidor sin escribir ningún script.

Otro objetivo importante de la tecnología JavaServer Faces es aprovechar los componentes familiares y los conceptos de nivel web sin limitarlo a una tecnología de scripting o lenguaje de marcado en particular. Las API de tecnología JavaServer Faces se colocan directamente sobre la API de Servlet

Esta combinación de API permite varios casos de uso de aplicaciones importantes, como el uso de diferentes tecnologías de presentación, la creación de sus propios componentes personalizados directamente desde las clases de componentes y la generación de resultados para varios dispositivos cliente.

La tecnología Facelets, disponible como parte de la tecnología JavaServer Faces, es la tecnología de presentación preferida para crear aplicaciones web basadas en la tecnología JavaServer Faces. Para obtener más información sobre las características de la tecnología Facelets, consulte el Capítulo 8, "Introducción a Facelets".

La tecnología de facelets ofrece varias ventajas.

El código puede ser reutilizado y extendido para los componentes a través de las funciones de los componentes de plantillas y compuestos.

Puede usar anotaciones para registrar automáticamente el bean administrado como un recurso disponible para las aplicaciones JavaServer Faces. Además, las reglas de navegación implícitas permiten a los desarrolladores configurar rápidamente la navegación de la página (consulte el Modelo de navegación para obtener más información). Estas características reducen el proceso de configuración manual para aplicaciones.

Lo más importante es que la tecnología JavaServer Faces proporciona una arquitectura rica para administrar el estado de los componentes, procesar los datos de los componentes, validar las entradas de los usuarios y manejar los eventos.



¿Qué es?

Beneficios

Manager Bean ◀

Scopes



ManagerBean

```
package com.tutorialspoint.test;

import javax.faces.bean.ManagedBean;

@ManagedBean(name = "helloWorld", eager = true)
public class HelloWorld {

    public HelloWorld() {
        System.out.println("HelloWorld started!");
    }

    public String getMessage() {
        return "Hello World!";
    }
}
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
    <head>
        <title>JSF Tutorial!</title>
    </head>

    <body>
        #{helloWorld.getMessage()}
    </body>
</html>
```

¿Qué es?

Beneficios

Manager Bean

Scopes



Scopes

@RequestScoped

Bean vive mientras que la solicitud-respuesta HTTP viva. Se crea en una solicitud HTTP y se destruye cuando finaliza la respuesta HTTP asociada con la solicitud HTTP.

@ViewScoped

Bean se mantiene mientras el usuario interactúe con la misma vista JSF en la ventana / pestaña del navegador. Se crea en una solicitud HTTP y se destruye una vez que el usuario devuelve a una vista diferente.

@SessionScoped

Bean vive mientras dure la sesión HTTP. Se crea en la primera solicitud HTTP que involucra a este bean en la sesión y se destruye cuando la sesión HTTP se invalida.

@ApplicationScoped

Bean vive mientras dure la aplicación web. Se crea en la primera solicitud HTTP que involucra a este bean en la aplicación (o cuando la aplicación web se inicia y el atributo `eager = true` se establece en `@ManagedBean`) y se destruye cuando la aplicación web se apaga.