CHRISTIAN DAVID NOREÑA VALENCIA 201255158

CUADRO RESUMEN LECTURAS 5 Y 6

<table>
<tr><td>¿Qué es un Managed Bean?</td><td colspan="2">Es una clase Bean de Java regular registrada con JSF. En otras palabras, es un bean de Java manejado por el framework JSF (Java Server Faces).</td></tr>
<tr><td>¿Por qué un Managed Bean?</td><td colspan="2">Los Managed Beans ofrecen un modelo de componente ligero alineado con el resto de la Plataforma Java EE. Para apoyar la inyección de recurso común y los servicios de ciclo de vida, los Managed Beans encajan bien dentro del modelo de programación de Java EE. Al mismo tiempo, su naturaleza ligera los hace un punto de partida natural para encapsular la funcionalidad de la aplicación, con el conocimiento de que ellos puedan ser transformados en componentes más poderosos si y cuando la necesidad ocurra. En este sentido, ellos pueden ser vistos como una versión mejorada del Java EE del modelo de componente JavaBeans encontrados en la plataforma Java SE.</td></tr>
<tr><td rowspan="2">Modelo básico de un Managed Bean</td><td>Definición del componente</td><td>• Declarado en la clase mediante la siguiente invocación: javax.annotation.ManagedBean.
• Un Managed Bean no puede ser una clase final, una clase abstracta, o una clase interna no estática.
• Un Managed Bean no puede ser serializado, diferente como sí puede serlo en un componente JavaBean regular.
• Las implementaciones de los Managed Bean pueden soportar Managed Beans que tengan un constructor sin argumentos.</td></tr>
<tr><td>Nombre</td><td>• Puede tener un nombre opcionalmente, un String. El nombre puede ser especificado mediante la anotación @ManagedBean. Ejemplo:
<code>@ManagedBean(“cart“)
public class ShoppingCart { ... }</code>
• Por cada Managed Bean nombrado, los componentes Java EE pueden hacer disponibles las siguientes entradas en JNDI, usando la misma nomenclatura usada para los componentes EJB:</td></tr>
</table>

		In the application namespace:
		`java:app/<module-name>/<bean-name>`
		
		In the module namespace of the module containing the Managed Bean:
		`java:module/<bean-name>`

	Ciclo de vida e inyección de recurso	javax.annotation.PostConstruct y javax.annotation.PreDestroy para identificar métodos de llamada de vuelta por el contenedor en los puntos apropiados en el ciclo de vida del bean.
	Manejo de hilos	Las invocaciones de métodos en un Managed Bean son ejecutados en el mismo hilo que el llamador.
	Interceptores	Un Managed Bean puede usar interceptores como se definen en la especificación del interceptor.
Anotaciones de alcance de los Managed Bean's	@RequestScoped	El bean vive tanto como la respuesta de petición HTTP vive. Éste se crea sobre una petición HTTP y éste se destruye cuando la respuesta HTTP asociada a una petición HTTP termina.
	@NoneScoped	El bean vive tanto como una evaluación de Lenguaje de Expresión (EL). Se crea sobre una evaluación EL y se destruye inmediatamente después de dicha evaluación.
	@ViewScoped	El bean vive tanto como el usuario interactúa con la misma vista JSF in la pestaña o ventana del navegador. Se crea sobre una solicitud HTTP y se destruye una vez el usuario realiza postback a una vista diferente.
	@SessionScoped	El bean vive tanto como la sesión HTTP vive. Se crea sobre la primera solicitud HTTP que involucra este bean en la sesión y se destruya cuando la sesión HTTP es inválida.
	@ApplicationScoped	El bean vive tanto como la aplicación web vive. Se crea sobre la primera solicitud que involucra este bean en la aplicación *orwhenthewebapplicationstartsupandtheeager = trueattributeissetin@ManagedBean* y se destruye cuando la aplicación web se apaga.
	@CustomScoped	El bean vive tanto como la entrada del bean en el custom Map el cual es creado para que este alcance viva.