Curso Spring – RichFaces – RichFaces y Spring Annotations

Marzo 2011 – JoeDayz
Susan Inga

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Contenido

- ✓ Introducción
- ✓ ¿Qué aprenderemos?
- ✓ Ganaremos experiencia en…
- ✓ Paso 1: Evaluando el estado inicial de la aplicación web
- ✓ Paso 2: Verificando la configuración de Spring
- ✓ Paso 3: Usando anotaciones
- ✓ Paso 4: Revisando el faces-config.xml
- ✓ Paso 5: Modificando el listado de cuentas
- ✓ Paso 6: Refactorizando la aplicación
- ✓ Paso 7: Refactorizando los detalles de las cuentas
- ✓ Paso 8: Configurando la navegabilidad

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Reward Dining

RichFaces y Spring Annotations

El taller Core Spring 3.0 de JoeDayz es
licenciado bajo Creative Commons
Attribution-Noncommercial-No Derivative
Works 3.0 United States License

Introducción

- ✓ En este laboratorio ganarás experiencia usando Spring para configurar la capa web de aplicación.
- ✓ Usarás Spring para configurar los beans de JSF como beans desplegados en el WebApplicationContext.
- ✓ Usarás la instalación ya existente y la transformarás usando la anotación @Component para configurar los beans que interactuan con la vista de la aplicación.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

¿Qué aprenderemos?

- ✓ La figura completa: ¿cómo Spring "ataca" en la arquitectura de una de una aplicación Java EE?
- ✓ ¿Cómo se usa Spring para organizar la configuración de archivos efectivamente?
- ✓ ¿Cómo usar anotaciones de inyección de dependencia de Spring tales como @ Component y @Autowired?
- ✓ Ventajas y desventajas de estas anotaciones.
- ✓ ¿Cómo se usa el Spring IDE para visualizar la configuración de la aplicación?

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Ganaremos experiencia en…

- ✓ @Autowired
- ✓ <context:component-scan/>
- ✓ @Component
- ✓ Sintaxis de la configuración XML Spring
- ✓ Spring IDE

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 1: Evaluando el estado inicial de la aplicación web

- ✓ Actualmente se cuenta con una aplicación que lista las cuentas registradas en el sistema. Adicionalmente, podemos agregar nuevas cuentas, actualizar datos y eliminarlas.
- ✓ Es decir, las funcionalidades ya se encuentran completamente implementadas.
- ✓ Ahora lo debemos hacer es realizar algunas configuraciones adicionales para hacer uso de anotaciones, de tal forma de disponer de todos los beans en el contenedor de Spring.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 2: Verificando la configuración de Spring

- Verifica el archivo de configuración de contexto en el archivo application-config.xml en la carpeta src/main/resources.

```
<bean id="accountRepository" class="accounts.internal.dao.HibernateAccountRepository">
    <constructor-arg ref="sessionFactory"/>
</bean>

<bean id="accountManager" class="accounts.internal.service.AccountManagerImpl">
    <property name="accountRepository" ref="accountRepository"/>
</bean>
```

- Observarás la una configuración en la que se hace necesario registrar los beans en el xml para que el contenedor de JSF pueda acceder a ellos usando EL en el faces-config.xml. Las borramos.

- Ahora usaremos solo anotaciones en toda la aplicación. Para ello haremos uso de la siguiente directiva:

```
<!-- Activates scanning of @Repository and @Service -->
<context:component-scan base-package="accounts"/>
```

- Con ello le diremos a Spring que vamos a trabajar con anotaciones.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 3: Usando anotaciones

- ✓ Ahora verifica el paquete accounts.internal.dao, el cual contiene la clase HibernateAccountRepository. Revísala.
- ✓ Como puedes observar, no está decorada con las anotaciones que le permitirán adquirir sus dependencias ni ser desplagada como bean en el contenedor.
- ✓ Procedemos a decorarla con las anotaciones correspondientes.
- ✓ Repetir los mismo pasos para la clase AccountManagerImpl del paquete accounts.internal.service.
- ✓ Finalmente, abrimos la clase AccountController y la decoramos con las anotaciones correspondientes.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 4: Revisando el faces-config.xml

- ✓ Ahora que nos inclinamos a usar el contenedor de Spring, debemos dejar de usar el de JSF.
- ✓ Para ello debemos abrir el archivo faces-config.xml :

```
<managed-bean>
    <description>account</description>
    <managed-bean-name>account</managed-bean-name>
    <managed-bean-class>accounts.web.AccountController</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
    <managed-property>
        <property-name>accountManager</property-name>
        <value>#{accountManager}</value>
    </managed-property>
</managed-bean>
```

- ✓ Como podemos observar la inyección se realizar vía la configuración de faces. Ahora ya no es necesario y eliminamos el contenido de <managed-bean/>.
- ✓ Sin embargo, debemos tomar en cuenta el scope puesto que también debemos decorar nuestro controller con el mismo scope elegido.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 5: Modificando el listado de cuentas

- Ahora que nuestro bean ya no se llama account (<managed-bean/>), sino que le hemos nombrado a nuestra conveniencia y le hemos asignado un ámbito:

```
@Component("accountForm")
@Scope("request")
public class AccountController extends BasePage {
```

- Debemos hacer el cambio en nuestro accountSummary.xhtml:

faces-config.xml | accountSummary.xhtml

```
<rich:dataTable value="#{accountForm.accounts}" var="item">
```

- Una vez terminado este cambio, levanta la aplicación y verifica si todo sigue funcionando correctamente.

- El listado es correcto pero las otras funcionalidades no.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 6: Refactorizando la aplicación

- Antes de continuar modificando el llamado a nuestro bean accountForm, debemos sugerir realizar una refactorización para separar el listado de cuentas, de las funcionalidades que nos permiten editar la información de cada cuenta.

- Se debe crear la clase AccountList, que no heredará de BasePage, y se debe implementar el método de listado de cuentas. Luego de ello debemos eliminarlo de AccountController.

```
@Component("accountList")
@Scope("session")
public class AccountList implements Serializable {
```

- Ahora si procedemos a realizar las llamadas correspondientes en los action de accountSummary.xhtml.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

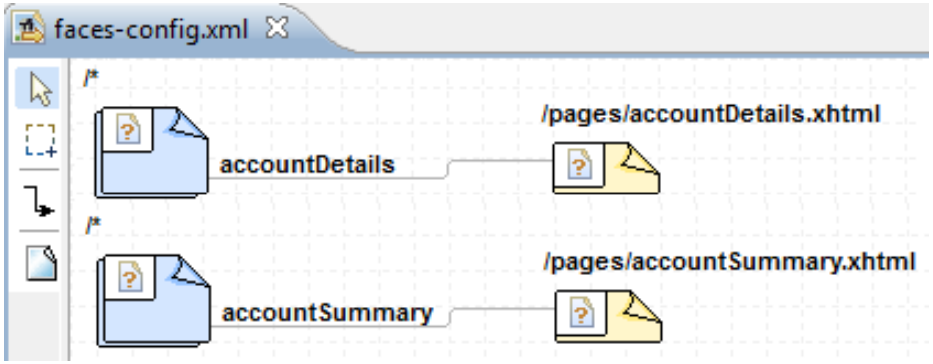
Paso 7: Refactorizando los detalles de las cuentas

- ✓ Al cambio de los nombres de los beans también hemos afectado a la vista accountDetails.xhtml.
- ✓ Debemos cambiar los llamados al bean account hacia el accountForm.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 8: Configurando la navegabilidad

- Actualmente la aplicación maneja la navegabilidad a nivel de llamados con nombres de vistas:

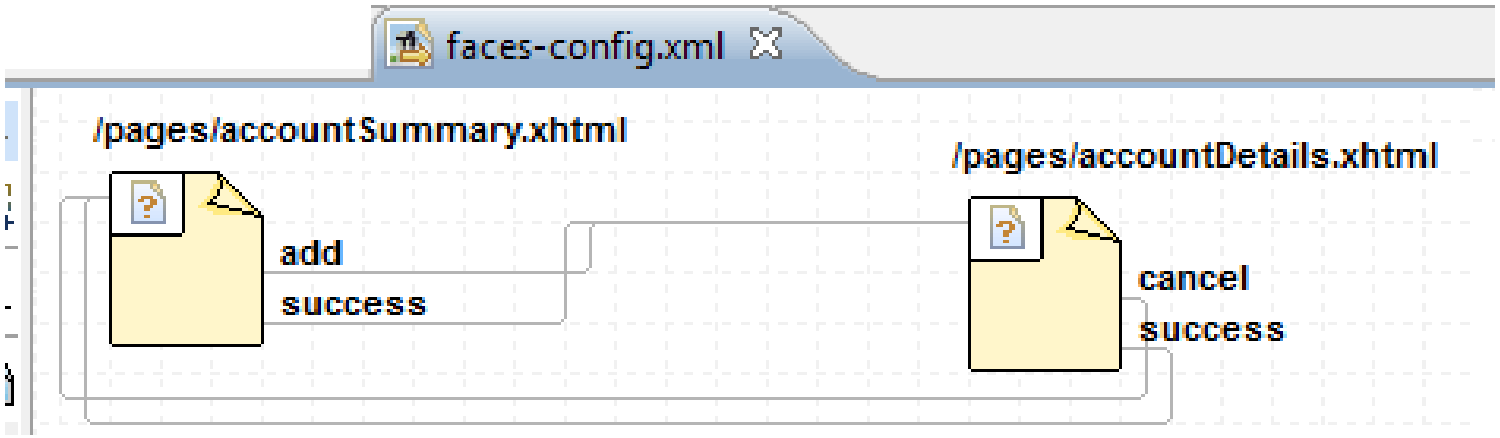


- Ahora, queremos usar algunas palabras claves para identificar las vistas que se deben renderizar según las funcionalidades que se elijan en la aplicación.

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Paso 8: Configurando la navegabilidad

- La configuración final que deseamos obtener es la siguiente:



- Asimismo, debemos realizar las modificaciones en los actions de las vistas y los valores de retorno de nuestros controladores.

- Ahora debemos probar la nueva implementación. Levanta la aplicación si todo está correcto, has terminado el lab!

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License

Enjoy it!

El taller Core Spring 3.0 de JoeDayz es licenciado bajo Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License