Servidores de aplicaciones

Índice

1 Introducción	2
2 Creación del dominio	2
3 Creación y configuración del EAR. Migración del proyecto	2

1. Introducción

En esta sesión tenemos dos objetivos principales:

- Por un lado, crear un dominio con weblogic, que será el que usaréis más adelante en los siguientes módulos, y que incorpore la mayor parte de las características vistas en el módulo de servidores de aplicaciones.
- Migrar el proyecto desarrollado hasta el momento a Weblogic, creando un EAR.

2. Creación del dominio

El dominio a crear debe tener las siguientes características:

Nombre del servidor	Dirección de escucha	Puerto de escucha	Nombre DNS	Grupo primario	Grupo secundario
admin		7001	Compañero1		
serva		4001	Compañero1	grupo1	grupo2
proxy		7001	Compañero2		
servb		4001	Compañero2	grupo2	grupo1
servc		5001	Compañero2	grupo2	grupo1

Dominio: Integracion

Debéis crear una fuente de datos de la base de datos del proyecto de integración. Llamaréis al recurso JNDI *biblioteca*.

En el servidor proxy podéis desplegar la aplicación creada en el ejercicio de la sesión 5 (tanto los nombres de los servidores como sus puertos de escucha son los mismos).

3. Creación y configuración del EAR. Migración del proyecto.

Ahora pretendemos crear un EAR (*Enterprise Application Resource*) con los proyectos desarrollados hasta el momento (el común y el web). Una aplicación enterprise (a menudo llamada simplemente aplicación, o aplicación J2EE) es una colección de recursos que se construyen juntos y se despliegan como una unidad a una instancia de un servidor Weblogic. También se pueden archivar en una fichero EAR.

Un proyecto EAR puede estar formado a su vez por varios proyectos: proyectos de utilidad, proyectos web, EJBs, servicios web, etc. El EAR nos va a permitir compartir recursos entre estos proyectos.

Vamos a comentar los pasos a seguir para crear el EAR. Partimos del proyecto de integración del proyecto web. Tenemos dos proyectos: el proy-int-comun y el proy-int-web.

- Importamos los dos proyectos dentro del Workshop. Nos aparecerán muchos errores, no tenemos que preocuparnos. Cambiamos el nombre de los proyectos a proy-int-comun2 y proy-int-web2. Nos puede dar un error indicando que no ha podido actualizar las referencias, no es problema.
- 2. Vamos a empezar primero por el proyecto común. Vamos a copiar el contenido del proyecto antiguo al nuevo. Seguimos los siguientes pasos:
 - 1. Creamos un nuevo proyecto de utilidad que llamaremos proy-int-comun.

🐨 New Project				X
Select a wizard Create a Utility project				
Wizards:				
CV3 CV3 CV3 CV3 CV3 CV3 CV3 CV3	oject n Project :t			
🔽 Show All Wizards.				Ŷ
	< Back	Next >	Finish	Cancel

Creación de un nuevo proyecto

2. Vamos dando las opciones por defecto hasta que nos permite seleccionar las *facets* que queremos que incluya. Deseleccionamos todas menos las indicadas en la siguiente figura.

W New Java Utility Module					
Select Project Facets Facets are units of functionality that car	n be added to a proj	ect.			
Presets: <custom></custom>				•	Save Delete
Project Facet U Java Utility Module Beehive Controls WebLogic Control Extension WebLogic Control Extension WebLogic Litting Module Exte	Version 5.0 1.0 5:5.0 5:9.2.0 on 9.2.0 n 9.2.0 n 9.2.0				
1					< Show Runtimes
		< Back	Next >	Fir	nish Cancel

Facets a seleccionar

3. Copiamos los fuentes (todos los paquetes de *src*) del antiguo al nuevo.

Aviso:

Si los seleccionamos con el ratón y los arrastramos estamos cambiando de sitio los ficheros. Si no los queremos perder, copiarlos y luego pegarlos en el destino.

- 4. Hacemos lo mismo con los directorios: test, resources y db.
- 5. Hemos terminado con el proyecto común. Tendremos algunos errores que luego solucionaremos.
- 3. Vamos a hacer lo mismo con el proyecto web. Seguimos estos pasos:
 - 1. Creamos un proyecto de web dinámica que llamaremos proy-int-web. Dejamos las *facets* indicadas:

W New Dynamic Web Project					X
Select Project Facets Facets are units of functionality that can	be added to a projec	t.			
Presets: <custom></custom>				•	Save Delete
Project Facet	Version				~
🛛 🔂 Dynamic Web Module	2.4				
Java	5.0				
Beehive Controls	1.0				
🗖 📄 Beehive NetUI	1.0				
🔄 🔲 📄 Java Annotation Processing S	15.0				
JSF	1.1				
DISTL	1.1				
Struts	1.2				
🗄 🗖 📄 Web Services					
- 🗖 📄 WebDoclet (XDoclet)	1.2.3				
🗌 📄 WebLogic Control Extensions	9.2.0				
🗌 🔲 📄 WebLogic Integrated Commor	19.2.0				_
🗌 🔲 📄 WebLogic Web App Extension	9.2.0				~
					<< Show Runtimes
		,			
		< Back	Next >	Fin	ish Cancel

Facets a seleccionar

- 2. Copiamos los fuentes y los directorios test, resources, cactus y logs.
- 3. Entramos dentro de WebContent. Copiamos los directorios css, imagenes, jsp y los ficheros index.jsp y jspRedirector.jsp.
- 4. Entramos dentro del WEB-INF. Copiamos los ficheros struts-config.xml, validation.xml, validator-rules.xml y web.xml. Sobreescribimos este último.
- 5. Copiamos las librerías del directorio WEB-INF/lib al correspondiente directorio del nuevo proyecto.
- 4. Pasamos ahora a crear el EAR. Seguimos estos pasos:
 - 1. Creamos un nuevo proyecto EAR.



Empezamos a crear un EAR

2. Lo llamaremos proy-int-ear. Le decimos que incluya los proyectos proy-int-comun y proy-int-web como módulos de esta aplicación. Nos dirá si incluye unas librerías, le decimos que sí.

🐨 New EAR Application Project	X
JZEE Modules to Add to the EAR Select the J2EE modules to add to the new EAR application from the list. Select New Module button to create a new J2EE module.	G
Proy-int-comun proy-int-web proy-int-web2	
Select All Deselect All New Module	
Content Directory: EarContent	
< <u>B</u> ack <u>M</u> ext > Einish	Cancel

Selección de módulos para el EAR

- 3. Copiamos las librerías que necesite el proyecto común del proy-int-web al directorio EarContent/APP-INF/lib del proyecto proy-int-ear.
- 5. Podemos borrar los proyectos antiguos, para que no nos molesten.
- 6. Nos vamos a las propiedades del proyecto común. Vamos a la opción *J2EE Module Dependencies* y seleccionamos las librerías que nos hagan falta. En esto momento tienen que haber desparecido todos los errores del proyecto de integración común.

🐨 Properties for proy-in	t-comun		
type filter text	J2EE Module Dependencies		() + () +
Info BeanInfo Path Builders File Templates J2EE Module Dependencie Java Build Path Java Code Style Java Compiler Javadoc Location JSP Decim Palette	This property page lets you setup the J2EI modules within an Enterprise Application to belong to the same Enterprise Application Enterprise application project name: proy Available dependent JARs: References to EJB JARs with EJB Client. O Use EJB JARs O Use EJB client JAF	E module dependencies with a All modules involved in the de on. -int-ear JARs 	other J2EE pendency need
- Profile Compliance and Va	JAR/Module	Project	~
Project Facets			-
Project References	APP-INE/lib/commons-lang-2.2		
Server	APP-INF/lib/commons-logging.jar		Up
- Task Tags	APP-INF/lib/commons-validator		
Validation	APP-INF/lib/displaytag-1.1.jar		Down
	🗖 🗍 APP-INF/lib/jakarta-oro.jar		Select All
····· XMLBeans	🗖 🛅 APP-INF/lib/jstl.jar		
	🗹 🦲 APP-INF/lib/log4j-1.2.14.jar		Deselect All
	🗹 🦲 APP-INF/lib/oscache-2.3.2.jar		
	ΔPP-INE/lib/standard_iar		<u>×</u>
	<u><</u>	<u>></u>	
	Manifest Class-Path:		
	APP-INF/lib/commons-lang-2:2: jar APP-IN APP-INF/lib/log4j-1:2:14: jar APP-INF/lib/o	F/lib/commons-logging.jar sscache-2.3.2.jar	
<		Restore Default:	s Apply
		OK	Cancel

Modificar las dependencias

7. Nos vamos a las propiedades del proyecto web. Seleccionamos todas las librerías y también, esto es muy importante, el proyecto común.

W Properties for prov-in	twee
Charlest of playing	
type filter text	J2EE Module Dependencies
- Info BeanInfo Path	J2EE Modules Web Libraries
 Builders File Templates J2EE J2EE Module Dependencie 	This property page lets you setup the J2EE module dependencies with other J2EE modules within an Enterprise Application. All modules involved in the dependency need to belong to the same Enterprise Application.
- Java Build Path ⊕ Java Code Style	Enterprise application project name: proy-int-ear
🛨 Java Compiler	Available dependent JARs:
Javadoc Location	References to EJB JARs with EJB Client JARs
···· JSP Design Palette ···· Profile Compliance and Va	C Use EJB JARs C Use EJB client JARs C Allow both
Project Facets	JAR/Module Project
Project References	APP-INF/lib/commons-v
Server	🗹 🗍 APP-INF/lib/displaytag
lask lags	🗹 🦲 APP-INF/lib/jakarta-oro
Valuation	🗹 🗍 APP-INF/lib/jstl.jar
XMI Beans	APP-INF/lib/log4j-1.2.1
hineboario	APP-INF/lib/oscache-2 Select All
	🗹 🦲 APP-INF/lib/standard.jar
	APP-INF/lib/struts-el.jar
	APP-INF/lib/strutstest-2
	🛛 🗹 🗍 proy-int-comun.jar proy-int-comun
	Manifest Class-Path:
	APP-INF/lib/anthr.jar APP-INF/lib/aspectirt-1.2.1.jar APP-INF/lib/cactus-1.7.2.jar APP-INF/lib/cactus-ant-1.7.2.jar APP-INF/lib/cargo-0.5.jar APP-INF/lib/commons-beautitis.jar APP-INF/lib/commons-collections-3.2.jar APP-INF/lib/commons-digester.jar APP-INF/lib/commons-fuelpolad.jar APP-INF/lib/commons-digester.jar APP-INF/lib/commons-ang-2.2.jar APP-INF/lib/commons-logging.jar APP-INF/lib/commons-valdator.jar APP-INF/lib/commons-logging.jar APP-INF/lib/commons-valdator.jar APP-INF/lib/commons-logging.jar APP-INF/lib/commons-valdator.jar APP-INF/lib/commons-logging.jar APP-INF/lib/jocatta-oro.jar APP-INF/lib/jistl.jar APP-INF/lib/log4j-1.2.14.jar APP-INF/lib/joscache-2.3.2.jar
<	Restore Defaults Apply
	OK Cancel

Modificar las dependencias

8. No tendría que dar errores. Si los da, haced un clean de todos los proyectos. Nuestra aplicación está preparada para desplegar.

Ahora tenemos que importar el servidor desde la solapa *Servers*. Le decimos New Server e importamos el dominio antes creado.



Crear un nuevo server

Con el servidor creado, llamamos a añadir y eliminar proyectos. Añadimos el EAR creado.



Añadir una aplicación

Aviso:

Acordaros de crear la fuente de datos en Weblogic y modificar la factoría de fuente de datos para que coja la conexión del JNDI del servidor. No es necesario indicar la dirección del servidor que tiene desplegada la fuente de datos, salvo que el despliegue de la aplicación sea en un servidor distinto.

Sólo queda poner en marcha el servidor (botón derecho y Start) y publicar la aplicación. Podemos modificar lo que queremos del proyecto y simplemente diciendo Republish nos desplegaría de nuevo la aplicación en el servidor.