

Desarrollo Basado en Componente y Servicios

Cómo crear un recurso JDBC en Glassfish

Nuestras aplicaciones JEE que acceden a la base de datos mediante ORM usando JPA, incluyen Entities (objetos a persistir) y DAOs (fachada para la gestión de las entities) creados, estos últimos, mediante EJBs de sesión sin estado. Para la gestión de las Entities se crea que lo que se denomina una *Unidad de Persistencia* (Persistence Unit). La *Unidad de Persistencia* se identifica mediante un nombre y se asocia a un recurso JDBC del servidor, que a la postre será el que realice la gestión de las operaciones CRUD. Esto se define en el fichero *persistence.xml* del componente EJB:

```
...  
<persistence-unit name="NombreUnidadPersistencia" transaction-type="JTA">  
  <jta-data-source>NombreRecursoJdbcServidor</jta-data-source>  
...
```

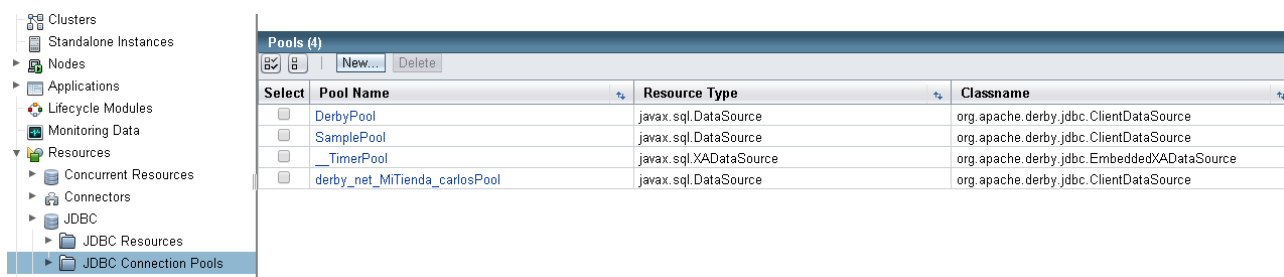
El nombre es usado para crear y acceder al gestor de persistencia (EntityManager) que es el encargado de realizar las operaciones CRUD.

Para que todo funcione el servidor tiene que tener definido el recurso (jta-data-source). Aquí vamos a ver cómo crear ese recurso mediante la consola de administración de Glassfish.

1. Creación del “pool” de conexiones JDBC

Un recurso JDBC es una “instancia” de algún elemento del “pool” de conexiones JDBC. Cada conexión contiene la información necesaria para que el servidor pueda conectarse con una base de datos. Vamos a describir los pasos a seguir para conectar con una base de datos creada en Derby.

Lo primero es seleccionar “JDBC Connection Pools” y en la ventana de la derecha, donde se listan los existentes, la opción “New”



La creación de un nuevo elemento implica dos pasos. En el primero hay que dar nombre al nuevo elemento (“Pool Name”) y definir el tipo de recurso (“Resource Type”) a *javax.sql.DataSource*. La opción “Database Driver Vendor” se puede dejar en blanco o seleccionar *Derby-30*.

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

General Settings

Pool Name: * EjCreacionPool

Resource Type: javax.sql.DataSource
Must be specified if the datasource class implements more than 1 of the interfaces

Database Driver Vendor: Derby-30
Select or enter a database driver vendor

Introspect: ☐ Enabled
If enabled, data source or driver implementation class names will enable introspection

Aquí vamos a seguir la opción de seleccionar el tipo de gestor de base de datos, así en el siguiente paso se rellenará automáticamente la opción “Datasource Classname” y aparecerán las propiedades a definir. Si se deja en blanco habrá que completar “a mano” esos campos, poniendo los valores de las figuras siguientes. En la parte de propiedades (“Additional Properties”) dar los valores adecuados a los campos “User”, “DatabaseName” y “Password”. Eliminar la propiedad “SecurityMechanism” para evitar problemas al conectar a la base de datos.

New JDBC Connection Pool (Step 2 of 2)

Identify the general settings for the connection pool. Datasource Classname or Driver Classname must be specified for the connection pool.

General Settings

Pool Name: EjCreacionPool

Resource Type: javax.sql.DataSource

Database Driver Vendor: Derby-30

Datasource Classname: org.apache.derby.jdbc.ClientDataSource
Select or enter vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:
Select or enter vendor-specific classname that implements the java.sql.Driver interface.

Ping: ☐ Enabled
When enabled, the pool is pinged during creation or reconfiguration to identify and warn of erroneous values for its attributes

Description:

Additional Properties (18)

Select	Name	Value
<input type="checkbox"/>	TraceFileAppend	false
<input checked="" type="checkbox"/>	SecurityMechanism	4
<input type="checkbox"/>	ConnectionAttributes	
<input type="checkbox"/>	Description	
<input type="checkbox"/>	TraceDirectory	
<input type="checkbox"/>	User	nombreUsuario
<input type="checkbox"/>	DatabaseName	nombreBaseDatosAConectar
<input type="checkbox"/>	Ssl	off
<input type="checkbox"/>	RetrieveMessageText	true
<input type="checkbox"/>	DataSourceName	
<input type="checkbox"/>	LoginTimeout	0
<input type="checkbox"/>	ShutdownDatabase	
<input type="checkbox"/>	TraceFile	
<input type="checkbox"/>	ServerName	localhost
<input type="checkbox"/>	CreateDatabase	
<input type="checkbox"/>	TraceLevel	-1
<input type="checkbox"/>	PortNumber	1527
<input type="checkbox"/>	Password	claveUsuario

Hacer clic en finalizar y ya habremos creado el elemento del “pool”.

Antes de continuar hay que probar que la conexión se ha realizado con éxito. Para ello seleccionar el elemento creado de los que aparecen en “JDBC Connection Pools”. Hacer Clic en “Ping”. Si sale el mensaje de la figura, la conexión creada ya está lista para ser usada.

General

Advanced

Additional Properties

Ping Succeeded

Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of database connections that are shared by multiple applications.

Load Defaults

Flush

Ping

General Settings

2. Crear el recurso

Una vez creado el elemento del “pool”, ya le podemos asociar recursos mediante los que acceder a la base de datos en las aplicaciones JEE.

Crear un nuevo recurso es más sencillo. Primero hacer en clic en “JDBC Resources” y luego seleccionar “New” en la parte de la derecha.



En la siguiente pantalla, dar un nombre al recurso (“JNDI Name”) y asociarle al “pool” creado.

New JDBC Resource
Specify a unique JNDI name that identifies the JDBC resource you want to create. The name must be unique within the same JNDI namespace.

JNDI Name: *

Pool Name: Click here to create new pools

Description:

Status:

derby_net_MiTienda_carlosPool

El nombre que pongamos al recurso será el que debe aparecer en “persistence.xml”:

```
...  
<jta-data-source>EjCreacionRecurso</jta-data-source>  
...
```