Ejercicios de introducción a Hibernate

Índice

Se os proporciona el fichero **hib-sesion1.doc** como plantilla para que contestéis a las preguntas formuladas en los ejercicios.

1. Ejercicio 1: Instalación y prueba de Hibernate

Vamos a instalar Hibernate en nuestro directorio de trabajo. Para ello se proporcionan dos ficheros (que también os podéis descargar desde la sección *Roadmap* de este módulo):

- **hibernate-3.1.5.tar.gz**: contiene las librerías necesarias para poder utilizar Hibernate.
- **hsqldb_1_8_0_1.zip**: contiene HSQLDB 1.8.0, es una base de datos relacional basada en java y que se almacena en memoria. La utilizaremos para hacer alguna prueba con Hibernate.

Para instalar y hacer una pequeña prueba del funcionamiento de Hibernate, seguiremos los siguientes pasos:

1. Descomprimir el fichero **hibernate-3.1.3.tar.gz** en nuestro directorio de trabajo. Esto creará el directorio `hibernate-3.1`.
2. Descomprimir el fichero **hsqldb_1_8_0_1.zip** en el directorio de trabajo. Esto creará el directorio `hsqldb_1_8_0_7`.
3. Para comprobar que Hibernate se ha instalado bien, vamos a ejecutar un ejemplo que viene con Hibernate, y que utiliza la base de datos HSQLDB. Para poder utilizar dicha base de datos solamente necesitamos copiar el driver JDBC (`hsqldb_1_8_0_7\hsqldb\lib\hsqldb.jar`) en el directorio **`hibernate-3.1\lib`**.
4. El directorio **`hibernate-3.1/etc`** contiene, entre otros:
   - El fichero de propiedades de Hibernate (**`hibernate.properties`**). Edítalo y comprueba que las propiedades de la base de datos a utilizar son las correspondientes a HSQLDB: simplemente hay que descomentar las propiedades referentes a dicha base de datos (y comentar las del resto).
   - Una plantilla de propiedades Hibernate (**`hibernate.properties.template`**). Aquí están todas las propiedades configurables de Hibernate, que podremos utilizar en nuestro propio proyecto, comentando y descomentando las propiedades que nos interesen.
5. Desde `hibernate-3.1` ejecutar: **`build eg`**. Con esto compilaremos y ejecutaremos el programa ejemplo que se encuentra en el directorio `hibernate-3.1/eg`. Verás que aparecen mensajes de información por pantalla al ejecutar la aplicación. Estos mensajes están precedidos por una indicación sobre el origen del mensaje, por ejemplo: `Configuration`, `Environment`, `SettingsFactory`, `HbmBinder` y `DriverManagerConnectionProvider`... Lee la información de pantalla e indica

en qué orden aparecen los cinco mensajes anteriores, así como una breve explicación de los pasos que sigue Hibernate según dichos mensajes.

6. Prueba efecto de la variable **`hibernate.show_sql`** poniendo su valor a `true` (en el fichero `hibernate.properties`). Vuelve a ejecutar el ejemplo e indica el formato del nuevo tipo de mensajes que aparecen en pantalla.
7. La variable anterior es una alternativa a la asignación **`org.hibernate.SQL = debug`** en el fichero `log4j.properties`. Compruébalo e indica el formato del nuevo tipo de mensaje.

2. Ejercicio 2: Hibernate desde Eclipse

Para probar las *Hibernate tools* vamos a utilizar el proyecto `hib-sesion1`, que se encuentra en el fichero **sesion01-ejercicios.zip**

Primero tenemos que instalar el *plugin* con las *Hibernate tools*, siguiendo las instrucciones que se indican en el último punto de esta sesión (apartado 1.12)

Una vez que hemos abierto el proyecto en Eclipse, realizaremos lo siguiente:

1. En las propiedades del proyecto, habilitamos Hibernate mediante: *Properties->Hibernate Settings -> Enable Hibernate 3 support*.
2. Añadimos los `*.jar` externos necesarios en *Properties->Java build path->Libraries* desde *hibernate-3.1/lib*. En el directorio `lib` del proyecto tenéis dicho conjunto mínimo de librerias. La librería `hsqldb.jar` está en *hsqldb/lib/*, y la librería `hibernate3.jar` está en el directorio *hibernate-3.1*.
3. Vamos a utilizar la base de datos HSQL, para lo cual crearemos un directorio en el proyecto llamado `log`, en donde la base de datos guardará los ficheros de datos.
4. Ahora creamos el fichero de configuración **`hibernate.cfg.xml`** en la carpeta **`src`** del proyecto, mediante *New->Other->Hibernate->Hibernate configuration file (cfg.xml)*, con los siguientes valores:
   - Dialecto de la base de datos: HSQL
   - Clase para el driver: org.hsqldb.jdbcDriver
   - Url de la conexión: jdbc:hsqldb:log/sesion01
   - Nombre de usuario para la conexión: sa

   Activaremos también la casilla *Create a console configuration*
5. En *Additional mapping files*, borramos el fichero *Event.hbm.xml*, ya que lo añadiremos en el fichero de configuración en el último paso.
6. En el *classpath*, para la consola de configuración añadiremos el fichero `jar` externo de la BD. Comprobamos que la consola de configuración creada (*hib-sesion1*) está asociada al proyecto en las propiedades del mismo: *Properties->Hibernate Settings -> Default Hibernate Console Configuration*.

7. Editamos el fichero de configuración *hibernate.cfg.xml* añadiendo las **propiedades**:
   - `connection.pool_size` con valor `1`. Vamos a utilizar el *pool de conexiones* que viene con Hibernate.
   - `show_sql` con valor `true`. Con esto permitimos mostrar por pantalla las sentencias SQL que se van ejecutando.
   - `current_session_context_class` con valor `thread`. Con esto habilitamos la gestión automática del contexto de las sesiones en Hibernate. Hablaremos de ello en la siguiente sesión.
   - `cache.provider_class` con valor `org.hibernate.cache.NoCacheProvider`. Con esto deshabilitamos el segundo nivel de *cache*. También hablaremos de ello en la última sesión.
   - `hbm2ddl.auto` con valor `create`. Con esto hacemos que se cree la base de datos de nuevo con cada ejecución del programa) (podéis consultar el fichero ***etc/hibernate.properties*** en la distribución de hibernate, para ver otras ocpiones posbiles para ésta y el resto de variables).
   - Finalmente añadimos el recurso de correspondencia (*mapping resource*) `Event.hbm.xml`.

Ya tenemos casi todo listo para ejecutar la aplicación. Nos queda (aunque esto es opcional) copiar el fichero *hibernate-3.1/etc/log4j.properties* en la carpeta `src`. Esto configura los mensajes de salida de Hibernate. Editamos dicho fichero para ver que hemos activado los mensajes de tipo **`info`** (variable `org.hibernate`). Podemos probar a ejecutar la aplicación con este mensaje activado y desactivado y ver las diferencias.

Para probar la aplicación cargamos el fichero `build.xml` y ejecutamos los *targets compile* y *run* en este orden.

Finalmente, realiza los siguientes **cambios** en el programa:

1. Prueba a cambiar la base de datos para utilizar MySQL en lugar de HQL. Para ello tienes que cambiar la configuración de la base de datos. La clase para el *driver* es *com.mysql.jdbc.Driver*, y el dialecto *org.hibernate.dialect.MySQLDialect*. Puedes consultar la plantilla de propiedades de Hibernate (*hibernate-3.1/etc/hibernate.properties.template*). Renombra el fichero de configuración actual como *hibernate.cfg.xml.hsqldb*, y crea otro fichero de configuración *hibernate.cfg.xml*. Cuando lo hayas creado haz una copia con el nombre *hibernate.cfg.xml.mysql*. De esta forma para cambiar de BD solamente habrá que reutilizar uno de los dos ficheros de configuración. Para utilizar MySQL podéis utilizar el usuario `root`, sin *password*, y la base de datos `hibernate`.
2. ¿Qué le ocurre a la BD entre dos ejecuciones consecutivas? ¿Por qué ocurre esto? Realiza los cambios necesarios para poder "ampliar" la base de datos con nuevos eventos en varias ejecuciones consecutivas.