

IES de l'Ebre

Persistència del Java Bean

MP6 UF4 Pt3

Marc Martí Mas

Curs 2020 - 2021

Tortosa, 18 de febrer de 2021

Índex

Índex	2
Java Bean	3
Vetoables Listeners	3
Property Change Listeners	5

Java Bean

Hem creat un Java Bean anomenat Connexió que serà l'encarregat de connectar-se amb la base de dades i executar consultes sobre ella.

Aquest Bean conté diferents propietats per a interactuar amb la base de dades:

```
private transient String propsDb;  
public static final String PROP_PROPSDB = "propsDb";
```

Propietat que passarà l'usuari amb el nom del fitxer de les propietats

```
private Connection conn = null;
```

Propietat per emmagatzemar la connexió quan es connecta amb la base de dades

```
private transient String update;  
public static final String PROP_UPDATE = "update";
```

Propietat que passarà l'usuari amb la sentència SQL per a fer canvis a la BBDD

```
private String select;  
public transient static final String PROP_SELECT = "select";
```

Propietat que passarà l'usuari amb la sentència SQL per a fer una consulta a la BBDD

```
private ResultSet result;
```

Propietat per emmagatzemar el resultat d'un select

```
private boolean close;  
public static final String PROP_CLOSE = "close";
```

Propietat que canviarà l'usuari per a tancar la connexió amb la base de dades

Vetoables Listeners

Per algunes de les propietats hem afegit uns listeners de tipus Vetoable (per a que es puguin llençar excepcions).

Per a la propietat de *propsDb* l'usuari hauria de passar el nom del fitxer que conté les propietats de la base de dades. Si tot ha anat bé, el programa buscarà la URL, usuari i contrasenya de la base de dades i s'intentarà connectar amb ella.

```

case PROP_PROPSDB:
    Properties properties = new Properties ();
    try {
        properties.load(new FileInputStream((String)
ev.getNewValue()));
        String url = properties.getProperty("url");
        String usuari = properties.getProperty("usuari");
        String contrasenya = properties.getProperty("contrasenya");
        conn = DriverManager.getConnection(url, usuari, contrasenya);

    } catch (Exception ex) {
        throw new PropertyVetoException("", ev);
    }

    break;

```

Execució del listener per a *propsDb*

La següent propietat està pensada per a que l'usuari passi una sentència SQL per a escriure sobre la base de dades. Es suposa que ja s'ha fet la connexió a la BBDD prèviament. Si la sentència és correcta el programa l'executarà contra la base de dades connectada.

```

case PROP_UPDATE:
    try {
        Statement stmt = conn.createStatement();
        stmt.executeUpdate((String) ev.getNewValue());
    } catch (SQLException ex) {
        throw new PropertyVetoException("", ev);
    }

    break;

```

Execució del listener per a *update*

L'última propietat a la que se li assigna un listener de tipus Vetoable és la de select. Aquesta és pareguda a l'anterior, però en comptes d'escriure només llegeix (fa selects). Per tant, retorna un resultat que emmagatzema a la propietat *result* que hem vist anteriorment. Aquí també es dona per fet que la connexió ha estat realitzada prèviament.

```

case PROP_SELECT:
    if (!((String) ev.getNewValue()).equals("")) {
        try {
            Statement stmt = conn.createStatement();
            result = stmt.executeQuery((String) ev.getNewValue());
        } catch (Exception ex) {
            throw new PropertyVetoException("", ev);
        }
    }

    break;

```

Execució del listener per a select

Property Change Listeners

Hem assignat els `PropertyChangeListener` a aquelles propietats que no ens interessa que llencin una excepció en cas d'error. En aquest cas només ens permet avisar que ha passat alguna cosa.

L'única propietat que hem assignat aquí és la de tancar la connexió. Quan el valor de la propietat canvia i té com a valor *true* el programa passa a tancar la connexió amb la base de dades que estava connectada.

```

case PROP_CLOSE:
    if ((boolean) ev.getNewValue() && conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {

        }
    }

    break;

```

Execució del listener per a close