

P2 – JDBC et architecture d'application – eVente (2)

Quelques composants de recherche et de sélection

Il est vraisemblable que de nombreux écrans de l'application eVente auront à offrir à l'utilisateur des possibilités de recherche et de sélection d'objets persistants. Nous allons développer des composants réutilisables offrant ces fonctionnalités.

INTRODUCTION

Nous allons prévoir des composants de recherche et de sélection pour les différents types d'objets que nous aurons à manipuler. Ces composants devront offrir un maximum de possibilités de recherche, des comportements analogues pour faciliter l'apprentissage de l'utilisateur. Nous veillerons à ce qu'ils occupent un espace minimum pour faciliter la composition des écrans.

COMPOSANTS DÉRIVÉS DE JCOMBOBOX

Les recherches-sélections sur des objets persistants présentant peu de valeurs distinctes et dont le seul critère pertinent de recherche pour l'utilisateur est un libellé peuvent se faire au travers d'une JComboBox.

Dans le cas de *eVente*, il semble que *Catégorie* et *Marque* satisfont à cette description.

Comme nous allons à créer des composants analogues pour *Catégorie* et *Marque* nous allons, dans un souci de généralisation, créer un composant générique `MaJComboBox` que nous pourrons utiliser non seulement pour *Marque* et *Catégorie* mais aussi pour d'autres objets dans d'autres applications. Ce composant sera placé dans un package d'outils utiles à différentes applications (p. ex. `be.esi.gxxxx.gui.outils`).

Ce composant devra présenter la liste des libellés des objets de type T (<T>), générer un `PropertyChangeEvent` en cas de modification de sélection, offrir un *getter* de l'objet sélectionné et un *setter*.

Nous vous fournissons cette classe `MaJComboBox` en annexe¹.

Une fois ce composant écrit nous pouvons en dériver un `JavaBean` `MaJComboBoxMarque` que nous placerons dans un package nommé `be.esi.gxxxxx.evente.gui.outils`.

```
import be.esi.adt.evente.dto.MarqueDto;
import be.esi.adt.gui.outils.MaJComboBox;
import java.util.Collection;

public class MaJComboBoxMarque extends MaJComboBox<MarqueDto> {
```

¹ Lisez-la maintenant.

```

public MaJComboBoxMarque() {
    super(new MarqueDto[0], true, "getId", "getLibelle");
}

public void setModel(Collection<MarqueDto> col) {
    MarqueDto[] marques=col.toArray(new MarqueDto[0]);
    super.setModel(marques);
}
}

```

EXERCICE 1

Après avoir inclu la classe `MaJComboBox` et écrit la classe `MaJComboBoxMarque`,

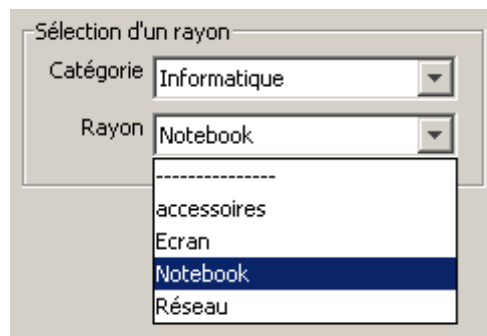
- x Testez les fonctionnalités de `MaJComboBoxMarque` dans une `JFrame`²
- x Ajouter la Javadoc nécessaire aux deux classes présentées ci-dessus
- x Réalisez une modification permettant d'obtenir la liste des marques ordonnées sur le libellé pour permettre un affichage ordonné.

Réalisez ensuite un composant `MaJComboBoxCategorie` analogue au précédent, un peu comme suit:



EXERCICE 2

Développez un composant plus complexe qui présentera deux `ComboBox` (évidemment dérivées de `MaJComboBox`). Le composant ne présentera que les rayons de la catégorie sélectionnée.



² Nous supposons à ce stade que vous avez bien terminé la première partie ... dans le cas contraire vous vous en rendez vite compte.