

Quelques composants de recherche et de sélection

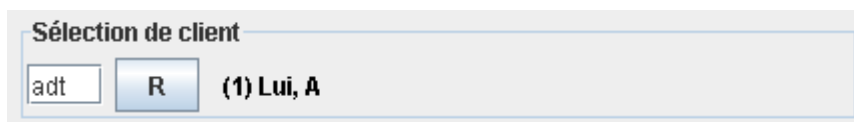
Il est vraisemblable que de nombreuses fonctionnalités de l'application eVente auront à présenter à l'utilisateur des possibilités de recherche. A nouveau, nous allons veiller à développer des composants réutilisables.

1 INTRODUCTION

Nous allons prévoir des composants 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 et devront occuper un espace minimum pour faciliter la composition de fenêtres tout en étant le plus efficace possible.

2 CLIENT

Ici, les critères de recherche peuvent être fort divers. Nous allons privilégier ceux qui nous semblent les plus souvent utilisés et qui déterminent un client (id et identifiant). Nous allons prévoir une zone de saisie (JTextField) de id ou de identifiant (nous pourrions discriminer l'un de l'autre par le fait que id doit être numérique et que identifiant doit comprendre au moins un caractère non numérique). Lorsque l'utilisateur presse 'Enter' ou que la zone perd le focus, nous lancerons la recherche; si elle aboutit, nous afficherons l'identifiant dans la zone de saisie et un texte de présentation du client trouvé dans un JLabel. Si elle n'aboutit pas, nous afficherons dans le JLabel un message adéquat. Ce composant devra offrir une méthode getClien() qui retournera le ClientDto sélectionné ou null si aucun client n'est sélectionné. De plus, il devra lancer un événement – limitons-nous à un PropertyChange - à chaque modification de sélection.



Sélection de client

adt R (1) Lui, A

ou



Sélection de client

R Client 'tre' inexistant

Malheureusement, ce composant ne permet pas, a priori, de réaliser une recherche complexe. Pour améliorer cela, il nous faut créer un JDialog, appellable à partir de la pression sur le bouton 'R', pour offrir des critères multiples de sélection. Si l'utilisateur ferme la fenêtre sans faire de sélection, l'affichage de notre composant ne sera pas modifié, si une sélection est faite, le composant affichera le client sélectionné.

Recherche de client

Critères de recherche

Début de nom: Début de Prénom:

eMail:

Actifs
 Tous Oui Non

Lancer la recherche

id	Uid	Nom	Prénom	Mail
2	Truc	Dupont	André	andre.dupont@truc.be
1	adt	Lui	A	

Sélectionner

3 PRODUIT

Développez un composant analogue pour la recherche et sélection d'un produit.

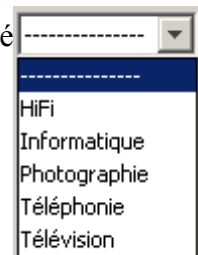
4 MARQUE ET CATÉGORIE

Nous supposons que Marque et Catégorie reprennent un nombre suffisamment faible d'objets que pour pouvoir permettre la sélection au travers d'une JComboBox (en pratique, ici, le seul critère de sélection censé pour l'utilisateur est le libellé).

Attention, pour que ce code soit opérationnel, il faut ajouter à `adt.evente.dto.Dto.java`:

```
public boolean equals(Object o){
    boolean b=false;
    if(o.getClass()!=this.getClass()) return b;
    else{
        if (this==o) return true;
        else {
            try {
                Method m = this.getClass().getMethod("getId");
                return m.invoke(this).equals(m.invoke(o));
            } catch (Exception ex){
                return false;
            }
        }
    }
}
```

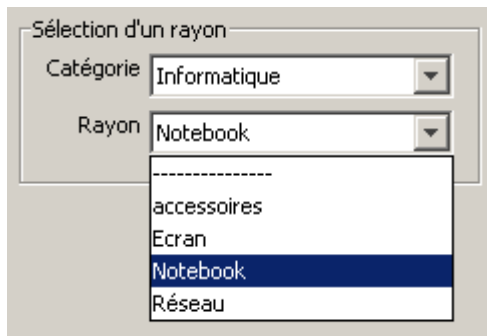
Un composant générique 'MaComboBox' vous est fourni ainsi qu'un dérivé 'ComboCatégorie':



Réalisez un composant analogue pour Marque.

5 RAYON

Dans ce cas, la sélection se fera au travers de deux 'MaComboBox'.



6 PRODUIT

Complétez le composant de recherche de produit pour offrir les critères de recherche de marque, catégorie et rayon au travers des composants qui viennent d'être développés.