

TD n°6 : Conception de logique applicative

Objectifs

- Concevoir des interfaces (vues)
- Créer leur logique comportementale

Situation initiale

- Créer le projet **koTd6** en important le fichier war dans eclipse : [kotd6.war](#)
- Démarrer le projet sur le serveur web (**Run as...**)
- afficher ensuite l'adresse : <http://127.0.0.1/kotd6/>

Conception des vues

Villes

Objectifs

Il s'agit de donner à la liste des villes le comportement suivant :

- Ajout de la sélection au clavier et à la souris
- Édition par défaut supprimée et remplacée par une édition directe (dans la liste)
- Ajout d'un bouton détail pour visualiser la liste des entreprises de la ville
- Ouvrir le template de la classe Kville :

Ajout de la sélection au clavier

la méthode **addSelector** permet d'ajouter le contrôle clavier sur la liste : le code de touche **113** correspond à la touche F2, et permet l'édition d'une ville : voir <http://tutorial.kobject.net/java/ajaxinclude/keyboard>

- Modifier le template des villes :

```
{#func:this.addSelector(113)#
{#func:this.setEditable(true)#
{#mask:<td>{cp}</td><td>{ville}</td>#
{#mask:<td>{cp}</td><td>{ville}</td>#
{#set:this.ajaxIncludes=true#}
{#set:this.listContentUrl="villes.do"#
{_ajax}
{_listContent}
{_page}
<div class="boxButtons">{_pageCounter}{_navBarre}</div>
{/_listContent}
```

- Tester la page **villes.do** et son comportement (clavier et souris)

| Action | Effet |
|--|--|
| F2 | Édition avec le formulaire de modification |
| Touches de direction Haut, bas, gauche, droite | Déplacement entre les villes |
| MAJ+Home | Atteindre la première page |
| MAJ+Fin | Atteindre la dernière page |
| MAJ+PageUp | Page précédente |
| MAJ+PageDown | Page suivante |
| Double clic | Déplacement sélection ligne |

The screenshot shows a web page titled 'Liste des villes'. At the top, there are two buttons: 'Liste des entreprises' and 'Liste des villes', with 'Liste des villes' being the active tab. Below the tabs is a table with 10 rows, each representing a city with its population and name. To the right of each city name is a 'Modifier' button. The rows are colored in a repeating pattern of light yellow and dark blue. The cities listed are: Caen, Authie, Clermont-Ferrand, Gap, La Ravoire, Bayeux, Authie Nord, Saint-Lô, Rennes, and Bordeaux. At the bottom of the table, there is a message 'Afficher 10 enregistrement(s) sur 26' and a navigation bar with icons for back, forward, and page numbers (1, 2, 3).

- Enlever l'édition de la liste

```
{#func:this.addSelector(113)#
{#func:this.setEditable(false)#
...

```

Activation de l'édition directe des membres

Pour qu'un membre soit éditabile, il faut qu'il appartienne à un élément DOM de la classe **editable**.

Nous allons ajouter un **Display** associé à la classe **KVille** pour modifier l'affichage des éléments de la liste : La méthode **showInList** à surdéfinir gère l'affichage de chaque membre de la liste.

```
package net.display;
import net.ko.displays.KObjectDisplay;
import net.ko.kobject.KObject;

public class VilleDisplay extends KObjectDisplay {
```

```

@Override
public String showInList(KObject ko, String memberName) {
    String result=super.showInList(ko, memberName);
    //si le nom du membre est cp ou ville, on le met dans un span de classe css
    //editable
    if("cp".equals(memberName) || "ville".equals(memberName)){
        result=<span class='editable' title='"+memberName+"'>"+result+"</span>";
    }
    return result;
}
}

```

- Associer ensuite le **Display** créé à la classe **KVille** dans le fichier **/conf/kox.xml** :

```

...
<class name="KVille" display="net.display.VilleDisplay">
    <member max="5" name="cp" required="1" type="string" />
    <member max="100" name="ville" required="1" type="string"
transform="onlyFirstWordUpper"/>
</class>
...

```

- Tester l'édition possible de la liste :

The screenshot shows a user interface for managing cities. At the top, there are two buttons: 'Liste des entreprises' and 'Liste des villes'. Below these, a table lists cities with their populations. The city 'Clermont-Ferrand' is highlighted with a blue background, and its name is in a white input field. A red hand cursor icon with the text 'double clic' is positioned next to the city name, indicating that a double click will trigger an action.

| Action | Effet |
|----------------|---------------------------------|
| Double clic | Mode édition du membre |
| ESCAPE | Sortie de l'édition |
| ENTREE | Sortie avec possible validation |
| Perte du focus | Sortie de l'édition |

A noter que l'édition n'est pour l'instant que visuelle, et ne permet pas la validation des modifications effectuées.

Ajout de la validation

- Editer le fichier **conf/mox.xml** et ajouter l'inclusion ajax suivante :
- Sur la page **villes.do**,
 - l'événement **updated** sur une zone de classe css **editable**,

- provoque le post vers une url virtuelle de validation (**updateVille.do**) de l'instance de **KVille**,
- et passe les informations contenues dans le champ édité (**target.innerHTML**) du champ (**target.title**)

target correspond à l'élément DOM qui a reçu l'événement :

```
<ajax-includes>
...
<request requestURL="villes.do">
  <js triggerSelector=".editable" triggerEvent="updated">
    <updateOne virtualURL="updateVille.do" operation="update"
kobjectShortClassName="KVille" targetId="info" method="POST">
      <field name="{js:target.title}" value="{js:target.innerHTML}"/>
    </updateOne>
  </js>
</request>
...
<ajax-includes>
```

- Ajouter dans le même fichier le mapping virtuel **updateVille.do**, associé à une action de type **updateOne** :

```
<mappings>
...
<virtualMapping requestURL="updateVille.do" mappingFor="updateOne"/>
...
</mappings>
```

- tester l'édition finalisée dans la page **villes.do** :

Message après modification et validation par **ENTREE** :



Ajout d'un champ supplémentaire dans la liste

Le champ supplémentaire devra afficher le nombre actuel d'entreprise(s) de la ville, dans un lien cliquable qui devra ensuite afficher la liste des entreprises de la ville.

- Modifier la classe **KVille** pour qu'elle charge automatiquement les entreprises depuis la base de données :

```
...
public KVille() {
  super();
 hasMany(KEntreprise.class);
```

```

    }
...

```

- Ajouter l'élément **btDetail** dans le masque de la liste (il s'agit d'un membre inexistant de la classe)

```

{#func:this.setEditable(false)#
{#func:this.addSelector(113)#
{#mask:<td>{cp}</td><td>{ville}</td><td>{btDetail}</td>#
{#mask:<td>{cp}</td><td>{ville}</td><td>{btDetail}</td>#
{#set:this.ajaxIncludes=true#
{#set:this.listContentUrl="villes.do"#
{_ajax}
{_listContent}
  {_page}
    <div class="boxButtons">{_pageCounter}{_navBarre}</div>
{/_listContent}

```

* Modifier le Display de la classe **KVille**, **VilleDisplay**, pour qu'il gère l'affichage de **btDetail** :

```

public class VilleDisplay extends KObjectDisplay {

    @Override
    public String showInList(KObject ko, String memberName) {
        String result=super.showInList(ko, memberName);

        if("cp".equals(memberName) || "ville".equals(memberName)){
            result=<span class='editable' title='
title='"+memberName+"'>"+result+"</span>";
        }
        if("btDetail".equals(memberName)){
            KVille ville=(KVille) ko;
            int nb=ville.getEnterprises().count();
            if(nb>0)
                result=<div><a id='alink-"+ville.getId()+"' class='default' title='Enterprises de
"+ville.getVille()+"'>"+KString.pluriel(ville.getEnterprises().count(),"entreprise"
)+"</a></div>";
            else
result=<div>"+KString.pluriel(ville.getEnterprises().count(),"entreprise")+"</div>
";
        }
        return result;
    }
}

```

Affichage des entreprises de la ville

- Ajouter une zone **divEnterprises** dans le template des villes qui recevra l'affichage des entreprises de la ville :

```

{#func:this.setEditable(false)#
{#func:this.addSelector(113)#
{#mask:<td>{cp}</td><td>{ville}</td><td>{btDetail}</td>#
{#mask:<td>{cp}</td><td>{ville}</td><td>{btDetail}</td>#
{#set:this.ajaxIncludes=true#}
{#set:this.listContentUrl="villes.do"#
{_ajax}
{_listContent}
  {_page}
    <div class="boxButtons">{_pageCounter}{_navBarre}</div>
{/_listContent}
<div id="divEntreprises"></div>

```

- Créer un nouveau template de type **list** pour afficher les entreprises de la ville.
- Associer le template à un Display **net.display.EntrepriseVilleDisplay** (à créer par la suite)
- **%ville%** permettra de récupérer la variable ville passée dans la requête

```

{#koDisplay:net.display.EntrepriseVilleDisplay#}
{#mask:<td>{rs}</td><td>{adresse}</td><td>{tel}</td>#
{#mask:<td>{rs}</td><td>{adresse}</td><td>{tel}</td>#
<fieldset>
<legend>%ville%</legend>
{_ajax}
{_listContent}
  {_page}
{/_listContent}
</fieldset>

```

Display

Le display va permettre de filtrer les entreprises de la liste, pour ne faire apparaître que les entreprises de la ville sélectionnée :

Surdéfinir la méthode **beforeLoading** de la façon suivante :

```

package net.display;

import javax.servlet.http.HttpServletRequest;

import net.ko.displays.KObjectDisplay;
import net.ko.http.objects.KRequest;
import net.ko.http.views.KPageList;
import net.ko.kobject.KObject;

public class EntrepriseVilleDisplay extends KObjectDisplay {

    @Override
    public void beforeLoading(Class<? extends KObject> clazz, KPageList list,
        HttpServletRequest request) {
        list.addWhere("idVille='"+KRequest.GET("idVille", request, "-1")+"'");
    }
}

```

{}

Mapping

* ajouter le mapping associé à la liste dans mox.xml

```
<mappings>
...
    <mapping requestURL="entrepParVille.do" responseURL="/WEB-INF/list/ville/entreprise.list"/>
...
</mappings>
```

Inclusion ajax

- ajouter l'inclusion sur le clic d'un lien **a** de classe css **default**, pour que chaque lien de ce type affiche **entrepParVille.do**, et lui passe les paramètres souhaités

```
<ajax-includes>
...
    <request requestURL="villes.do">
        ...
        <js triggerSelector="a.default" triggerEvent="click">
            <include targetURL="entrepParVille.do"
targetParams="idVille={js:$vId(target.id)},ville={js:target.title}"
targetId="divEntreprises"></include>
        </js>
    </request>
</ajax-includes>
```

- Tester le comportement de la vue villes.do (la touche F2 ou un clic sur le lien d'une ville sélectionnée provoque l'affichage des entreprises de la ville):

Liste des entreprises
Liste des villes

| | | |
|-------|------------------|---------------------|
| 14000 | Caen | 4 entreprises |
| 14000 | Authie | Aucun(e) entreprise |
| 63000 | Clermont-Ferrand | 1 entreprise |
| 5000 | Gap | 1 entreprise |
| 73490 | La Ravoire | 1 entreprise |
| 14400 | Bayeux | 3 entreprises |
| 14100 | Authie Nord | Aucun(e) entreprise |
| 50000 | Saint-Lô | Aucun(e) entreprise |
| 35000 | Rennes | Aucun(e) entreprise |
| 33000 | Bordeaux | Aucun(e) entreprise |

Afficher 10 enregistrement(s) sur 26

1 2 3 4

Entreprises de Caen

| | | |
|---------------------------|-----------------------------|----------------|
| Marie Automobiles | 36, Boulevard André Detolle | 02.31.22.16.36 |
| Garage Crapart | 26 Boulevard Maréchal Juin | 02.31.95.55.15 |
| S.A.R.L Lenrouilly-Vivier | 35, Avenue Henry Chéron | 08.99.18.75.66 |
| Garage Colbert | 6 Rue Jean-Baptiste Colbert | 08.99.02.88.69 |

Entreprises

Sur la liste des entreprises :

1. Activer les touches clavier (sans permettre l'édition)
2. Modifier la classe **KEntreprise** pour qu'elle charge automatiquement ses évaluations
3. Créer un Display **net.display.EntrepriseDisplay** pour afficher les évaluations dans la liste (méthode `showInList` à surdéfinir)
 1. la colonne Évaluations doit faire apparaître la liste des évaluations (leur date uniquement)
 2. Un clic sur 1 évaluation doit faire apparaître dans une boîte de dialogue l'évaluation cliquée. Il faudra à cet effet :
 - Créer un template de type **show** pour la classe `evaluation`
 - Utiliser l'inclusion ajax **includeDialog** pour afficher ce template dans une boîte de dialogue.

Concevoir l'interface suivante :

Liste des entreprises | Liste des villes

Rechercher une entreprise
Rechercher...

| Raison sociale | Adresse | ville | Evaluations |
|--------------------------------------|-------------------------------------|------------------------|-------------------|
| Concessionnaire PEUGEOT ABCIS Centre | 27 avenue du Brezet | 63000 Clermont-Ferrand | Evaluations (1) ▾ |
| | | | • 08/12/2013 |
| Centre auto Feu Vert Gap | 93 Avenue d'Embrun | 5000 Gap | Aucune(0) |
| Marie Automobiles | 36, Boulevard André Detolle | 14000 Caen | Evaluations (1) ▾ |
| Concessionnaire FIAT GMSA | RN6 - Route de Challes | 73490 La Ravoire | Aucune(0) |
| Garage Crapart | 26 Boulevard Maréchal Juin | 14000 Caen | Aucune(0) |
| S.A.R.L Lenrouilly-Vivier | 35, Avenue Henry Chéron | 14000 Caen | Aucune(0) |
| Garage Colbert | 6 Rue Jean-Baptiste Colbert | 14000 Caen | Aucune(0) |
| Mary Automobiles Bayeux | Avenue Yves Dubois de la Cotardière | 14400 Bayeux | Aucune(0) |
| Cgé Bodin | Route de Caen | 14400 Bayeux | Aucune(0) |
| Heudier Automobiles | Route du Molay-Littry | 14400 Bayeux | Aucune(0) |

Afficher 10 enregistrement(s) sur 10 [Navigation icons]

Entreprise

Raison sociale
Concessionnaire PE...

Mail
[non renseigné]

Tél.
04.73.98.03.06

Adresse
27 avenue du Brezet

Evaluation

Evaluation

Date
08/12/2013

Mail
myaddressmail@g...

Détails par critère

| Critère | Commentaire | Note |
|---------|-------------------|------|
| Accueil | Aucun | 12 |
| SAV | Sans commentaires | 5 |

Fermer

Afficher la liste des entreprises

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/javaee/td6?rev=1386786362>

Last update: **2019/08/31 14:42**

