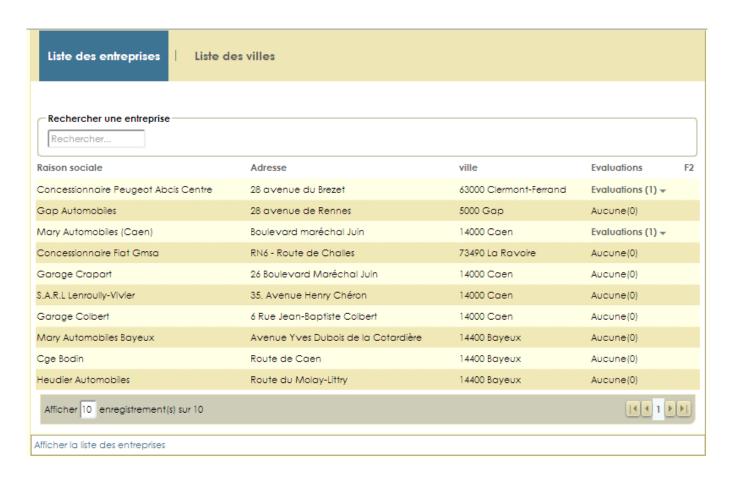
2025/10/16 17:35 1/14 Entreprises

Entreprises

-- Objectifs

Conception d'une interface complète, en utilisant mappings, inclusions ajax, templates et displays.

-- Liste des entreprises

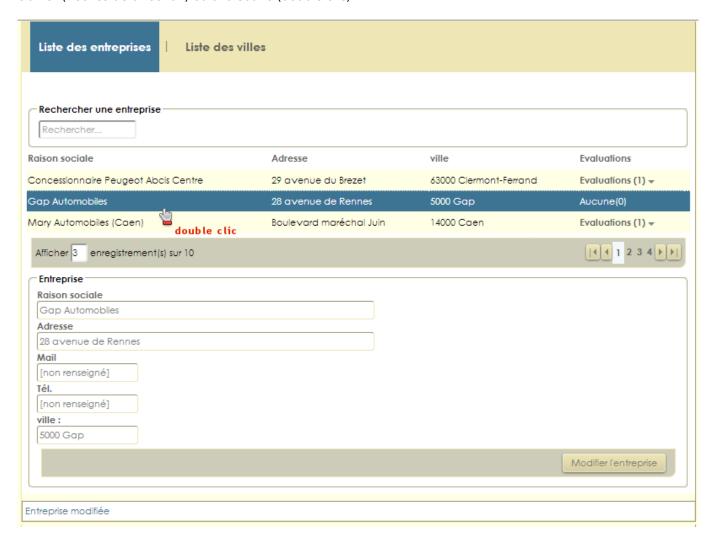


```
{#func:this.setEditable(false)#}
{#func:this.addSelector(113)#}
{#func:this.setFiltreCaption("Rechercher une entreprise")#}
{#set:this.ajaxIncludes=true#}
{#set:this.isShowCaption=true#}
{#set:this.listContentUrl="entreprises.do"#}
{\#mask:\{rs\}{adresse}{ville}{evaluations}{F2}}
}#}
{#mask:{rs}{adresse}{ville}{evaluations}{F2
}#}
{_ajx}
{_listContent}
   { filtre}
   {_page}
   <div class="boxButtons">{_pageCounter}{_navBarre}</div>
{/_listContent}
```

<div id="detail" style="display:none"></div>

-- Affichage des informations détaillées

Comportement attendu : La zone DOM d'id **detail** va afficher les informations sur l'entreprise sélectionnée au clavier (flèches de direction) ou à la souris (double clic).



-- Création de la vue en lecture seule

 Créer la vue destinée à afficher les détails sur une entreprise avec un template d'affichage d'objet (*.show) :

```
{#set:this.ajaxIncludes=true#}
{_fieldset}
<div id="divEntrepriseShow">
{rs}{adresse}{mail}{tel}{ville}
<div class="boxButtons"><a class="btn" id="btUpdateEntreprise">Modifier
l'entreprise</a></div>
</div>
<div id="divEntrepriseUpdate" style="display: none">
</div>
{/div>
{/_fieldset}
```

2025/10/16 17:35 3/14 Entreprises

</div>

-- Ajout de la dynamique

Ajouter un mapping vers le fichier show :

```
<mappings>
    ...
    <mapping requestURL="entrepDetail.do" responseURL="WEB-
INF/forms/entreprise.show"/>
    ...
    </mappings>
```

- Ajouter le chargement de la page entrepDetail.do dans la div detail (pour l'instant masquée), sur toute requête commençant par entreprises et se terminant par .do (Expression régulière entreprises{#(.*?)#}.do)
- La condition !\$('divEntrepriseUpdate') évite que la page ne soit chargée plusieurs fois (divEntrepriseUpdate est une div existante de la page entrepDetail.do)

- La page **entrepDetail.do** est mise à jour sur changement de l'entreprise active dans la liste (**#list-KEntreprise.onItemChange**)
- Le rafraichissement des infos est réalisé à partir d'une inclusion refreshFormValues, qui permet de mettre à jour les données (sans changer ni charger à nouveau l'interface) via une réponse JSON.
- e.detail.value permet de récupérer la clé primaire de l'entreprise active (sur la sélection)
- Les deux div (divEntrepriseShow et detail) contenant les informations sont affichées

• L'URL virtuelle changeEntrep.do doit être déclarée dans les mappings :

```
<mappings>
    ...
    <virtualMapping requestURL="changeEntrep.do"
mappingFor="refreshFormValues"/>
    ...
    </mappings>
```

Tester la page des entreprises, et l'affichage des informations de détail sur changement de la sélection (en bleu)

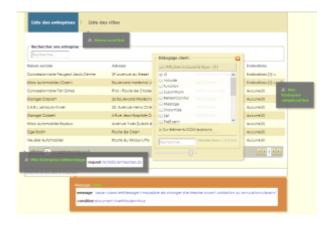
Débogage éventuel

Mode debug

Utiliser le mode debug, si le résultat n'est pas conforme à celui attendu :

- mettre debug=true dans le fichier config.ko,
- redémarrer l'application (avec classes.main),
- rafraîchir la page concernée :

Il est alors possible de visualiser les inclusions ajax chargées, et leurs dépendances.



Site map

L'adresse http://127.0.0.1:8080/kotd6/siteMap.main permet également de visualiser la logique des inclusions ajax :

2025/10/16 17:35 5/14 Entreprises



-- Corrections sur l'affichage de la vue

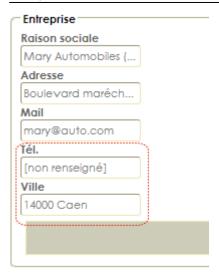
Nous allons maintenant compléter l'affichage de la vue en agissant sur la classe display (net.display.EntrepriseDisplay) associée à la classe KEntreprise :

- Il s'agit :
 - o d'afficher la mention [non renseigné] si l'un des membres contient une valeur nulle ou vide
 - o de compléter l'information sur le membre ville non affiché

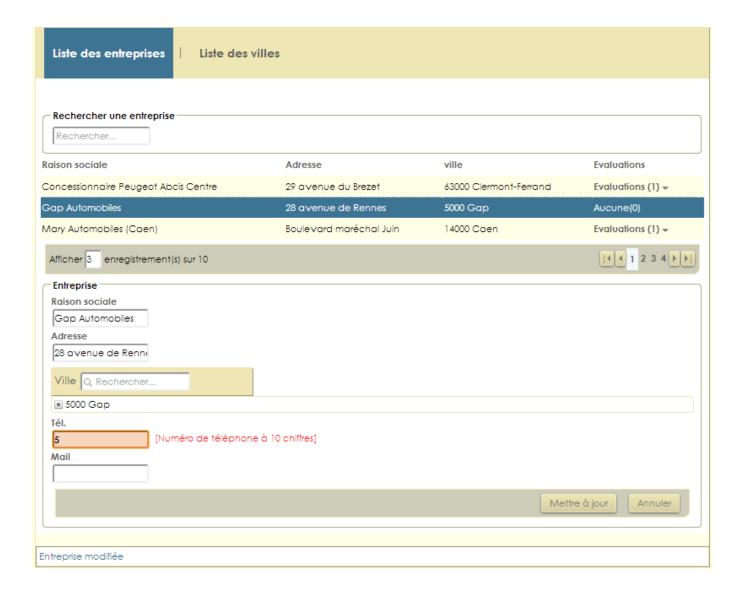
```
public class EntrepriseDisplay extends KObjectDisplay {
    @Override
    public String getRefreshValue(KObject ko, String memberName, KObjectController
koc, HttpServletRequest request) {
        String result=super.getRefreshValue(ko, memberName, koc, request);
        if("ville".equals(memberName)){
            KEntreprise entreprise=(KEntreprise) ko;
            if(entreprise.getVille()!=null)
                result=entreprise.getVille()+"";
        if(KString.isNull(result))
            result="[non renseigné]";
        return result;
    }
    @Override
    public String[] getRefreshFields() {
        return new String[]{"ville"};
}
```

Tester les modifications





-- Formulaire de modification d'entreprise



-- Création du formulaire

Créer le formulaire entreprise.view à partir d'un template (*.view) dans WEB-INF/forms

2025/10/16 17:35 7/14 Entreprises

-- Modification des contrôles

• Faire quelques ajustements dans le fichier **conf/kox.xml** sur les contrôles de validité et l'affichage par défaut d'une instance de **KEntreprise** dans un formulaire :

- La raison sociale de l'entreprise est transformée avant insertion dans la base :
 - 1ère lettre des mots en majuscules (onlyFirstWordUpper)
 - Suppression des espaces avant et après (trim)
 - o prévention contre les attaques xss : (xssClean)
- le champ idVille est affiché dans un contrôle de type radioAjaxList
- les champs **tel** et **idVille** deviennent obligatoires (**required="1"**)
- Modifier la méthode **tel** du transformer de la classe pour qu'elle gère bien **tous les cas possibles** sur la saisie d'un numéro :

```
    0231101112 ⇒ 02.31.10.11.12
    02-31-10-11-12 ⇒ 02.31.10.11.12
    02 31 10 11 12 ⇒ 02.31.10.11.12
```

```
@SuppressWarnings("serial")
public class MyTransformers extends KTransformer {
    public static String tel(String value){
       value=value.replaceAll("(\\d{2}).?(\\d{2}).?(\\d{2}).?(\\d{2}).?(\\d{2})",
"$1\\.$2\\.$3\\.$4\\.$5");
    return value;
```

```
}
}
```

-- Ajout de la dynamique

-- Mappings

• Ajouter les mappings suivants :

- entreprise.do correspond au formulaire
- submitFormKEntreprise.do correspond à l'url virtuelle en charge de la validation du formulaire
- entreprisesRefresh.do permet le rafraîchissement de la liste des entreprises après validation

-- Inclusions ajax

Formulaire et validation

• Ajouter les inclusions ajax dans mox.xml permettant l'affichage et la validation :

```
<ajax-includes>
        <request requestURL="entrepDetail.do">
            <js triggerSelector="#btUpdateEntreprise" triggerEvent="click">
                <showHide targetSelector="#divEntrepriseShow" visible="0"/>
                <showHide targetSelector="#divEntrepriseUpdate" visible="1"/>
                <includeForm targetId="divEntrepriseUpdate"</pre>
targetParams="id={js:$('list-KEntreprise').selector.getValue()}"
formKobjectShortClassName="KEntreprise" timeout="5000"
                targetURL="entreprise.do" formTargetId="divEntrepriseUpdate"
formName="frmEntrep" formButtonId="btUpdateEntrep" formButtonKeyCode="13">
                    <include targetURL="entreprisesRefresh.do"</pre>
targetId=" ajxContent"/>
                    <message targetId="info">'Entreprise modifiée'</message>
                </includeForm>
            </js>
        </request>
   </ajax-includes>
```

2025/10/16 17:35 9/14 Entreprises

- * Dans la vue entrepDetail.do, un clic sur le bouton btUpdateEntreprise :
 - masque la div **divEntrepriseShow** (vue en lecture seule)
 - affiche la div divEntrepriseUpdate (destinée à l'affichage du formulaire)
 - intègre l'inclusion includeForm permettant l'affichage du formulaire entreprise.do dans la div divEntrepriseUpdate et sa validation complète :
 - le formulaire reçoit l'id de l'entreprise à modifier (id de l'élément sélectionné dans la liste) :
 targetParams="id={js:\$('list-KEntreprise').selector.getValue()}"
 - le post du formulaire (**frmEntrep**) sera affiché dans la div **divEntrepriseUpdate** sur le clic du bouton **btUpdateEntrep** ou sur la frappe de la touche ENTREE (code 13)
 - Sur validation du formulaire :
 - La liste des entreprises est actualisée entreprisesRefresh.do vers la div _ajxContent (interne à la liste)
 - Le message d'information 'Entreprise modifiée' est affiché dans la div info
 - le message de validation est affiché 5 secondes (timeout="5000")

Annulation

• Ajouter les inclusions ajax dans **mox.xml** permettant l'annulation :

- Dans la vue **entreprise.do**, un clic sur le bouton **btCancelEntreprise** ou la frappe de la touche ESCAPE (code 27) :
 - Vide la div divEntrepriseUpdate
 - Affiche la div divEntrepriseShow
 - Affiche le message 'Modifications annulées' dans la div info

Tester le comportement implémenté, sans oublier de redémarrer l'application (à partir de **classes.main**)

-- Ajustements

Pour terminer, il s'agit d'empêcher le déplacement de la sélection dans la liste, lorsqu'une entreprise est en cours de modification dans le formulaire :

Ajout d'une variable dans le document

sur modification d'une entreprise (entreprise éditée, et l'un des champs a été modifié, le membre

document.insertMode de la page prend la valeur true.

• le mode insertion passe à false si l'édition est annulée

• ou si l'édition est validée :

```
<ajax-includes>
        <request requestURL="entrepDetail.do">
            <js triggerSelector="#btUpdateEntreprise">
                <showHide targetSelector="#divEntrepriseShow" visible="0"/>
                <showHide targetSelector="#divEntrepriseUpdate" visible="1"/>
                <includeForm targetId="divEntrepriseUpdate"</pre>
targetParams="id={js:$('list-KEntreprise').selector.getValue()}"
formKobjectShortClassName="KEntreprise" timeout="5000"
                targetURL="entreprise.do" transition="opacityShow"
formTargetId="divEntrepriseUpdate" formName="frmEntrep"
formButtonId="btUpdateEntrep" formButtonKeyCode="13">
                    <include targetURL="entreprisesRefresh.do"</pre>
targetId="_ajxContent"/>
                    <message targetId="info">'Entreprise modifiée'</message>
                    <function script="document.insertMode=false;"/>
                </includeForm>
            </is>
        </request>
   </ajax-includes>
```

- si insertMode vaut true, on empêche le déplacement de la sélection dans la liste des entreprises en annulant l'événement **itemchange** (**e.preventDefault()**)
- on affiche un message dans la zone info

```
<ajax-includes>
...
    <request requestURL="entreprises{#(.*?)#}.do">
```

2025/10/16 17:35 11/14 Entreprises

```
<js triggerSelector="#list-KEntreprise" triggerEvent="itemchange" >
                <function script="if(document.insertMode) e.preventDefault();"/>
                <refreshFormValues keyValues="{js:e.detail.value}"
kobjectShortClassName="KEntreprise" virtualURL="changeEntrep.do"
condition="!document.insertMode">
                    <showHide targetSelector="#detail" visible="1"/>
                    <showHide targetSelector="#divEntrepriseShow" visible="1"/>
                    <message targetId="divEntrepriseUpdate">''</message>
                    <function script="document.insertMode=false;"/>
                </refreshFormValues>
                <message targetId="info"</pre>
condition="document.insertMode==true"><![CDATA['<span</pre>
class=\'errMessage\'>Impossible de changer d\'entreprise avant validation ou
annulation</span>']]></message>
            </js>
        </request>
   </ajax-includes>
```

Tester le comportement implémenté, sans oublier de redémarrer l'application (à partir de **classes.main**)

-- Action par défaut

Dans une liste avec sélection (comme celle des entreprises), la touche **F2** du clavier permet de déclencher le clic sur un élément de la ligne sélectionnée, dont la classe css est **default**. Nous allons ajouter cet élément, et l'associer à l'édition de l'entreprise sélectionnée :

• Vérifier que le masque du template entreprise.list contient bien un champ **F2** (on aurait pu lui donner un non quelconque totalement différent)

```
{#func:this.setEditable(false)#}
{#func:this.addSelector(113)#}
{#func:this.setFiltreCaption("Rechercher une entreprise")#}
{#set:this.ajaxIncludes=true#}
{#set:this.isShowCaption=true#}
{#set:this.listContentUrl="entreprises.do"#}
{\text{sh}(-td}(rs){adresse}{ville}{evaluations}<{r2}</td>
{\#mask:\{rs\}{adresse}{ville}{evaluations}{F2}}
}#}
{ ajx}
{ listContent}
   {_filtre}
   { page}
   <div class="boxButtons">{ pageCounter}{ navBarre}</div>
{/ listContent}
<div id="detail" style="display:none"></div>
```

- Surdéfinir les méthodes suivantes de EntrepriseDisplay :
 - o getCaption pour modifier l'étiquette (en-tête de colonne) du champ F2 (le rendre invisible)

Last update: 2019/08/31 14:42

• showInlist pour faire en sorte que le champ F2 soit un span de classe css default

```
public class EntrepriseDisplay extends KObjectDisplay {
    @Override
    public String getCaption(KObject ko, String memberName) {
        String result=super.getCaption(ko, memberName);
        if(memberName.equals("F2")){
            result="";
        }
        return result;
    }
    @Override
    public String showInList(KObject ko, String memberName) {
        String result= super.showInList(ko, memberName);
        if(memberName.equals("F2")){
            result="<span class='default'></span>";
        }
        return result;
    }
}
```

• Ajouter le comportement sur les éléments de la classe css **default** : le clic sur un élément de la classe **default** provoque le clic sur le bouton **updateEntreprise**, et édite donc l'entreprise sélectionnée.

Tester le comportement implémenté, et notamment le bon fonctionnement de la touche **F2** du clavier, sans oublier au préalable le redémarrage de l'application (à partir de **classes.main**)

-- Boîte de message et suppression

Il s'agit maintenant de permettre la suppression d'une entreprise, sur frappe de la touche **SUPPR** du clavier (code 46), en ayant au préalable un message de confirmation.

```
<ajax-includes>
```

2025/10/16 17:35 13/14 Entreprises

```
<request requestURL="entreprises{#(.*?)#}.do">
            <js triggerSelector="document" triggerEvent="keyup" keyCode="46">
                <messageDialog title="Suppression d'entreprise"</pre>
transition="rotation10">
                     <button caption="Oui">
                         <deleteOne virtualURL="deleteEntrep.do"</pre>
kobjectShortClassName="KEntreprise" targetId=" ajx">
                             <include targetURL="entreprisesRefresh.do"</pre>
targetId=" ajxContent"/>
                         </deleteOne>
                         <function script="return true;"/>
                    </button>
                     <button caption="Annuler">
                         <function script="return true;"/>
                    </button>
                     <message><![CDATA['Supprimer l\'entreprise</pre>
<b>'+Forms.Utils.getFirstChild($('list-
KEntreprise').selector.getSelectedElement(),'td').innerHTML+'</b> ?'
                     ]]></message>
                </messageDialog>
            </js>
        </request>
    </ajax-includes>
```

- L'inclusion ajax **messageDialog** permet d'afficher une boîte de message,
- munie dans le cas présent de 2 boutons :
 - le bouton **"Oui"** confirme la suppression :
 - il fait appel à l'inclusion **deleteOne**, qui supprime l'entreprise sélectionnée dans la liste
 - puis rafraîchit la liste des entreprises
 - et ferme enfin la boîte de dialogue (return false;)
 - le bouton "Annuler" ferme la boîte de dialogue

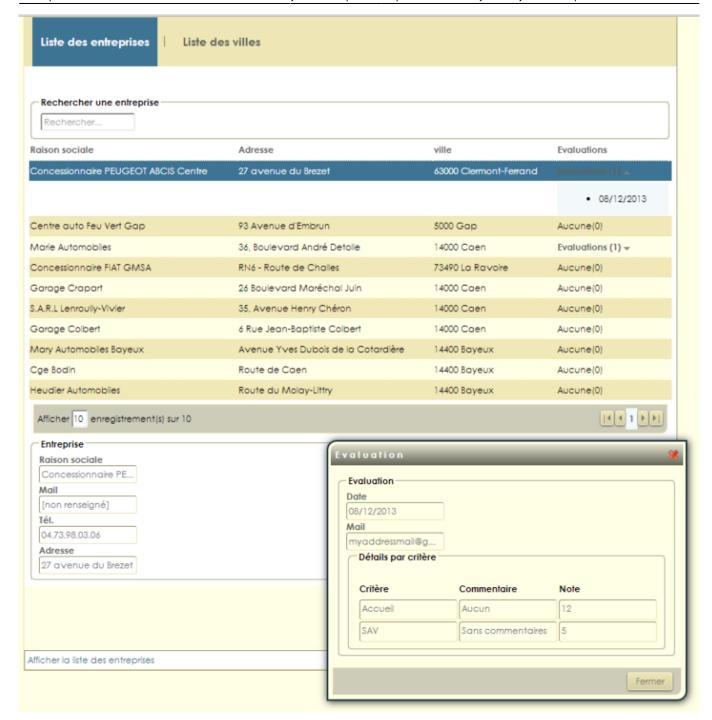
-- Affichage des évaluations

Sur la liste des entreprises :

- 1. Modifier la classe **KEntreprise** pour qu'elle charge automatiquement ses évaluations
- Modifier le Display net.display.EntrepriseDisplay pour afficher les évaluations dans la liste (méthode showInList à modifier)
 - 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 :





From:

http://slamwiki2.kobject.net/ - SlamWiki 2.1

Permanent link:

http://slamwiki2.kobject.net/javaee/td6/partie2?rev=1386893866

Last update: 2019/08/31 14:42

