

Université Lille1

Wiki Home	Download Grouper	Grouper Guides	Community Contributions	Developer Resources	Deployment Guide
---------------------------	----------------------------------	--------------------------------	---	-------------------------------------	----------------------------------

This information related to version 1.6.3 of Grouper. These pages are written in French, I will summarize here the titles :

I. Adapting LiteUI

1. Franchify interface
2. Do not display the link to AdminUI + reduce details about the group and members
3. Display the type of person (student, staff, teachers, etc.)
4. Change the import / export menu "More choices ..."
 - 4.1 Import
 - 4.2 Export

I. Adaptation de LiteUI

A noter : J'ai dû rentrer dans le code de LiteUI fin août car je n'obtenais pas correctement les accents ce qui était bloquant pour la mise en production (normalement, résolu depuis : [Internationalisation Grouper 1.6](#)). Du coup, j'ai utilisé les connaissances acquises sur le fonctionnement du code LiteUI pour apporter les modifications rapides que je souhaitais. Je vous les livre, elles ne suivent pas forcément ce qui est écrit sur le wiki Grouper : [Customiser Grouper UI](#) - Si un autre établissement peut apporter des rectifications/conseils/éclairages, ils sont les bienvenus !

Attention, problème avec IE9 pour le paging : je vais donner l'astuce de cliquer sur le bouton "affichage de compatibilité" juste après la saisie de l'url, à côté de la loupe.



1. Franciser l'interface

J'ai francisé l'interface en modifiant le fichier [chemin d'installation de Grouper-UI]/conf/resources/grouper/nav.properties : toutes les clés simpleMembershipUpdate (je n'ai francisé que LiteUI car AdminUI est vraiment orienté administrateur Grouper donc par des informaticiens). Voir le fichier joint : listesModificationsFrancaises.txt Il faut intégrer ces clés dans votre fichier nav.properties et commenter celles correspondantes en anglais. --- Ou mieux : créer un fichier nav.properties français avec déjà ces clés, voir comment il peut surcharger celui en anglais (où le mettre pour qu'il soit pris en compte ?) --

Attention au tri avec les accents, j'ai également modifié un source pour faire le tri correctement, voir sur la liste grouper-users nov 2011 (modif du SubjectSortWrapper.java)

2. Ne pas afficher le lien vers AdminUI + réduire les détails sur le groupe et les membres

ne pas afficher le lien vers AdminUI : modifier la jsp simpleMembershipUpdateMain.jsp (dans ../grouper-ui/webapp/WEB-INF/grouperUi/templates) en supprimant :

```
<grouper:message valueTooltip="${simpleMembershipUpdateContainer.text.viewInAdminUiTooltip}" value="${simpleMembershipUpdateContainer.text.viewInAdminUi}" />
```

Pour réduire les détails sur le groupe, utiliser les propriétés du fichier media.properties, par exemple pour Lille1 :

```

#if the id row should show on the screen by default
simpleMembershipUpdate.showIdRowByDefault=false

#if the id path row should show on the screen by default
simpleMembershipUpdate.showIdPathRowByDefault=false

#if the alternate id path row should show on the screen by default
simpleMembershipUpdate.showAlternateIdPathRowByDefault=false

#if the uuid row should show on the screen by default
simpleMembershipUpdate.showUuidRowByDefault=false

```

pour ne pas afficher tous les détails sur les personnes (notamment l'uid qui peut être une donnée sensible), modifier simpleMembershipUpdateSubjectDetails.jsp :



```

...
<!-- this is a map of key value pairs Map.Entry -->
<c:forEach var="attribute" items="${simpleMembershipUpdateContainer.subjectDetails}">
    <!-- this could be misleading, but put in some common labels from nav.properties and tooltips

    subject.summary.displayName=Path
    subject.summary.extension=ID
    subject.summary.createTime=Created
    subject.summary.createSubjectId=Creator ID (entity ID)
    subject.summary.createSubjectType=Creator entity type
    subject.summary.modifyTime=Last edited
    subject.summary.modifySubjectId=Last editor ID (entity ID)
    subject.summary.modifySubjectType=Last editor entity type
    subject.summary.subjectType=Entity type

    -->
    <c:set var="subjectSummaryNavKey" value="subject.summary.${attribute.key}" />
    <!-- change some common ones so they dont overlap -->
    <c:if test="${simpleMembershipUpdateContainer.subjectForDetails.type.name == 'group'}">
        <c:set var="subjectSummaryNavKey" value="subject.summary.group.${attribute.key}" />
        <c:set var="lacleG" value="${attribute.key}" />
        <c:set var="afficheG" value="0" />
    </c:if>
    <c:choose>
        <c:when test="${! empty navNullMap[subjectSummaryNavKey]}">
            <c:if test="{lacleG == 'displayExtension'}">
                <tr>
                    <td>
                        <c:out value="Nom" />
                        <c:set var="afficheG" value="1" />
                    </td>
                </tr>
            </c:if>
            <c:if test="{lacleG == 'extension'}">
                <tr>
                    <td>
                        <c:out value="Nom court" />
                        <c:set var="afficheG" value="1" />
                    </td>
                </tr>
            </c:if>
            <c:if test="{lacleG == 'displayName'}">
                <tr>
                    <td>
                        <c:out value="Adresse longue" />
                        <c:set var="afficheG" value="1" />
                    </td>
                </tr>
            </c:if>
            <c:if test="{lacleG == 'name'}">

```

```

        <tr>
        <td>
            <c:out value="Adresse courte" />
            <c:set var="afficheG" value="1" />
        </td>
    </c:if>
</c:when>
<c:otherwise>
    <c:set var="lacle" value="\${attribute.key}" />
    <c:set var="affiche" value="0" />
    <c:if test="\${lacle == 'screenLabel'}">
        <tr>
        <td>
            <c:out value="Nom" />
            <c:set var="affiche" value="1" />
        </td>
    </c:if>
    <c:if test="\${lacle == 'mail'}">
        <tr>
        <td>
            <c:out value="Courriel" />
            <c:set var="affiche" value="1" />
        </td>
    </c:if>
    <c:if test="\${lacle == 'supannAffectation'}">
        <tr>
        <td>
            <c:out value="Affectation ou Formation" />
            <c:set var="affiche" value="1" />
        </td>
    </c:if>
    <c:if test="\${lacle == 'supannCivillite'}">
        <tr>
        <td>
            <c:out value="Etat civil" />
            <c:set var="affiche" value="1" />
        </td>
    </c:if>
    </c:otherwise>
</c:choose>
<c:if test="\${affiche == '1'}">
    <td>
        <c:out value="\${attribute.value}" />
    </td>
</tr>
</c:if>
<c:if test="\${! empty navNullMap[subjectSummaryNavKey]}">
    <c:if test="\${afficheG == '1'}">
        <td>
            <c:out value="\${attribute.value}" />
        </td>
    </tr>
</c:if>
</c:if>
</c:forEach>
</table>
</div>
</div>
</div>
<!-- End: simpleMembershipUpdateSubjectDetails.jsp -->

```

3. Afficher le type de personne (étudiant, personnel, enseignant, etc)

J'ai modifié la méthode "convertSubjectToLabel" dans le fichier GrouperUiUtils.java (sous .../java/src/.../grouper/ui/util) :

```

public static String convertSubjectToLabel(Subject subject) {
    String label = null;
    // ajout bw
    String typeMember = null;
    String typeMemberF = null;
    HashMap franciserType = new HashMap();
    franciserType.put("student", "étudiant");
    franciserType.put("faculty", "enseignant");
    franciserType.put("employee", "personnel");
    franciserType.put("affiliate", "affilié");
    franciserType.put("researcher", "chercheur");
    franciserType.put("retired", "retraité");
    franciserType.put("emeritus", "émérite");

    ....

    typeMember = subject.getAttributeValue("eduPersonPrimaryAffiliation");
    if (franciserType.get(typeMember)==null) {
        typeMemberF = typeMember;
    }
    else {
        typeMemberF = franciserType.get(typeMember).toString();
    }
    if (!StringUtils.isBlank(typeMember)) {
        label = label + " - " + typeMemberF;
    }
    return label;
}

```

Ne pas oublier de déclarer l'attribut concerné dans le sources.xml côté grouper-API : ici, eduPersonPrimaryAffiliation

Nota : si vous souhaitez également avoir cette information dans AdminUI, il faut modifier la jsp : subjectView.jsp ainsi que le fichier media.properties de Grouper-UI pour déclarer la valeur de subject.display.bw.lille\ldap (ligne subject.display.bw.lille1\ldap=eduPersonPrimaryAffiliation dans media.properties), "lille1:ldap" représentant la source déclarée dans le fichier sources.xml de Grouper-API :

```

<tiles:importAttribute ignore="true"/><c:set var="attrKey2" value="*subject.display.bw.${viewObject.source.id}"
/><c:set var="attrKey" value="subject.display.default"/><c:set var="attrKeyGroup" value="subject.display.g"/>
<%--<tiles:importAttribute ignore="true"/><c:set var="attrKey2" value="*subject.display.bw.${viewObject.source.
id}"/><c:set var="attrKey" value="*subject.display.${viewObject.source.id}"/><c:if test="${empty mediaMap
[attrKey]}"><c:set var="attrKey" value="subject.display.default"/></c:if> --%>
<c:if test="${viewObject.isGroup}"><grouper:tooltip key="group.icon.tooltip"/>
    <c:out value="${viewObject[mediaMap[attrKeyGroup]]}" /></c:if><c:if test="${empty inLink}"
><span${viewObject.subjectType}"/>Subject"></c:if><c:out value="${viewObject[mediaMap[attrKey]]}" /> [<c:out
value="${viewObject[mediaMap[attrKey2]]}" /></span></c:if>

```

4. Modifier l'import/export du menu "Plus de choix..."

4.1 l'import

Je voulais que l'import se base sur les courriels universitaires.. J'ai donc modifié dans sources.xml le searchSubjectByIdentifiant en indiquant rechercher sur l'attribut mail :

```

<search>
  <searchType>searchSubjectByIdentifier</searchType>
  <param>
    <param-name>filter</param-name>
    <param-value>
      (&amp; (mail=%TERM%) (objectClass=supannPerson))
    </param-value>
  </param>
  <param>
etc

```

J'ai également masqué la demande de la source pour l'import car par défaut, à Lille1, nous n'avons que le ldap en sources de subject : dans simpleMembershipUpdateImport.jsp, j'ai commenté la partie concernée :

```

<%-- <tr>
  <td><grouper:message value="\${simpleMembershipUpdateContainer.text.importAvailableSourceIds}" /><
/td>
  <td><%-- \${fn:escapeXml(simpleMembershipUpdateContainer.sourceIds) }
  <table>
    <tr>
      <th>sourceId</th><th>Source name</th>
    </tr>
    <c:forEach items="\${contextContainer.sources}" var="source">
      <tr>
        <td><c:out value="\${source.id}" escapeXml="true"/></td>
        <td><c:out value="\${source.name }" escapeXml="true"/></td>
      </tr>
    </c:forEach>
  </table>
</td>
</tr> --%>

```

4.2 l'export

Je ne souhaitais plus afficher l'export des identifiants (information sensible). J'ai commenté la partie concernée dans le fichier SimpleMembershipUpdateMenu.java dans .../java/src/.../grouperUI/serviceLogic) :

```

/** else if (StringUtils.equals(menuItemId, "exportSubjectIds")) {
  guiResponseJs.addAction(GuiScreenAction.newAlertFromJsp(
    "/WEB-INF/grouperUi/templates/simpleMembershipUpdate/simpleMembershipUpdateExportSubjectIds.jsp"));
} **/
else if (StringUtils.equals(menuItemId, "exportAll")) {
  guiResponseJs.addAction(GuiScreenAction.newAlertFromJsp(
    "/WEB-INF/grouperUi/templates/simpleMembershipUpdate/simpleMembershipUpdateExportAll.jsp"));
} else if (StringUtils.equals(menuItemId, "import")) {
  guiResponseJs.addAction(GuiScreenAction.newDialogFromJsp(
    "/WEB-INF/grouperUi/templates/simpleMembershipUpdate/simpleMembershipUpdateImport.jsp"));
} else {
  throw new RuntimeException("Unexpected menu id: " + menuItemId + "");
}

```

et je n'exporte que le nom et le mail, à modifier dans media.properties :

```

#simpleMembershipUpdate.exportAllSubjectFields=sourceId, screenLabel, entityId, name, description
simpleMembershipUpdate.exportAllSubjectFields=name, mail

```

Voir sur la liste grouper-users nov 2011 comment exporter un attribut ldap multivalué (modif de SimpleMembershipUpdateImportExport.java mais sera mis dans les prochaines versions).