Grouper UI v2.3 developers guide

This page shows information for Grouper developers (or perhaps customizers) to work with the new Grouper v2.2 UI.

Naming

Give each JSP a unique name. e.g. dont put attributeAssignAddValue.jsp in various folders. Helps with patching

Common operations

Externalized text in JSP

```
${textContainer.text['groupLabelPath']}
```

Escape HTML in JSP (ampersand, lessthan, greater than, double quote)

```
${grouper:escapeHtml(grouperRequestContainer.groupContainer.guiGroup.pathColonSpaceSeparated)}
```

Escape Javascript in JSP (HTML and single quotes)

```
${grouper:escapeJavascript(someVar.someField)}
```

sdf

Validation errors

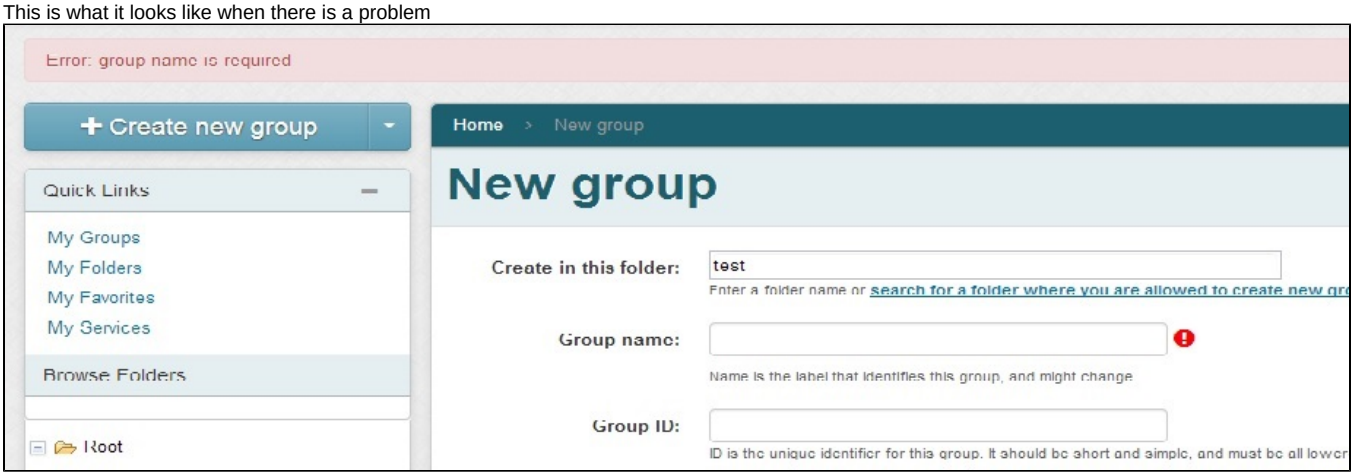
You can put icons next to fields where there are validation errors. Just make sure there is a way to refer to where you want the icons to go, give a message type and message.

```
      if (StringUtils.isBlank(displayExtension)) {

        guiResponseJs.addAction(GuiScreenAction.newValidationMessage(GuiMessageType.error,
            "#groupName",
            TextContainer.retrieveFromRequest().getText().get("groupCreateErrorDisplayExtensionRequired")));
        return;

      }
```

In this case in the JSP the textfield has an id of groupName

```
                    <label for="groupName">${textContainer.text['groupCreateNameLabel'] }</label>
                    <div>
                      <input type="text" id="groupName" name="displayExtension" />
                    </div>
```

This is what it looks like when there is a problem



sfd

Hide / show

Here is an example of a hide/show. Use CSS classes and jquery. Set display none on initially hidden things

```
<p class="shownAdvancedProperties"><a href="#"
  onclick="$('.hiddenAdvancedProperties').show('slow'); $('.shownAdvancedProperties').hide('slow'); return
false;"
  >${textContainer.text['groupCreateAdvanced'] } <i class="icon-angle-down"></i></a></p>
<p class="hiddenAdvancedProperties" style="display: none"
  onclick="$('.hiddenAdvancedProperties').hide('slow'); $('.shownAdvancedProperties').show('slow'); return
false;"
  ><a href="#" >${textContainer.text['groupCreateHideAdvanced'] } <i class="icon-angle-up"></i></a></p>
<div class="hiddenAdvancedProperties" style="display: none">
```

Call ajax

There are two ways to call ajax, one is ajax that affects the page but does not change the URL (for bookmarks and back button). This is used when submitting a form, or when doing minor things like sorting or paging.

```
<a href="#" onclick="ajax('../app/UiV2Group.addMemberSearch?groupId=${grouperRequestContainer.groupContainer.
guiGroup.group.id', {formIds: 'addMemberSearchFormId'}); return false;" >Search</a>
```

The operation must be in the edu.internet2.middleware.grouper.grouperUi.serviceLogic package. The method must have the signature:

```
public void addMemberSearch(HttpServletRequest request, HttpServletResponse response) {
```

The other way to call ajax is to affect the URL, you would do this when you want the request to be bookmarkable and the back button to work. These are GETs and dont submit forms, everything should be in the URL

```
<a href="#" onclick="return guiV2link('operation=UiV2Group.viewGroup&groupId=${grouperRequestContainer.
commonRequestContainer.guiGroup.group.id}');" />
```

Note, if you are in a screen, and submit it, and dont do a guiv2link, then the url will be what it was before, and if someone clicks on the same url, it wont change (also you want bookmark and refresh to work). For this reason, if you want to show another screen on submit, you should tell the browser to go there, like this:

```
guiResponseJs.addAction(GuiScreenAction.newScript("guiV2link('operation=UiV2Subject.viewSubject&memberId=" +
member.getId() + "');"));
```

sfd

Search form (with modal results)

We have search forms which are similar to the comboboxes, but might have more options, and might be easier to use on some devices

Put a div like this in the screen near the top (e.g. groupHeader.jsp):

```
                    <div id="member-search" tabindex="-1" role="dialog" aria-labelledby="member-search-label"
aria-hidden="true">
                      <div><a href="#" data-dismiss="modal" aria-hidden="true">x</a>
                        <h3 id="member-search-label">Search for an entity</h3>
                      </div>
                      <div>
                        <form id="addMemberSearchFormId">
                          <input name="addMemberSubjectSearch" type="text" placeholder="Search for an entity"/>
                          <button onclick="ajax('../app/UiV2Group.addMemberSearch?
groupId=${grouperRequestContainer.groupContainer.guiGroup.group.id}', {formIds: 'addMemberSearchFormId'});
return false;" >Search</button>
                        </form>
                        <div id="addMemberResults">
                        </div>
                      </div>
                      <div>
                        <button data-dismiss="modal" aria-hidden="true">Close</button>
                      </div>
                    </div>
```

Open the modal window with a button below the combobox

```
Enter an entity name or ID, or <a href="#member-search" onclick="$('#addMemberResults').empty();" role="button"
data-toggle="modal" style="text-decoration: underline !important;">search for an entity</a>.
```

Have an include for the results that can be controlled via ajax

```
<%@ include file="../assetsJsp/commonTaglib.jsp"%>

                          <table>
                            <thead>
                              <tr>
                                <th>Entity Name</th>
                              </tr>
                            </thead>
                            <tbody>

                              <c:forEach items="${grouperRequestContainer.groupContainer.guiSubjectsAddMember}"
                                var="guiSubject" >

                                <tr>
                                 <td><a href="#" data-dismiss="modal">${guiSubject.
screenLabelShort2noLinkWithIcon }</a></td>
                                </tr>


                              </c:forEach>
                            </tbody>
                          </table>
```

Write the controller in Java

```
  /**
   * search for a subject to add to the group
   * @param request
   * @param response
   */
  public void addMemberSearch(HttpServletRequest request, HttpServletResponse response) {

    GrouperRequestContainer grouperRequestContainer = GrouperRequestContainer.retrieveFromRequestOrCreate();

    final Subject loggedInSubject = GrouperUiFilter.retrieveSubjectLoggedIn();
    GuiResponseJs guiResponseJs = GuiResponseJs.retrieveGuiResponseJs();

    GrouperSession grouperSession = null;

    try {
      grouperSession = GrouperSession.start(loggedInSubject);

      Group group = retrieveGroupHelper(request, AccessPrivilege.UPDATE).getGroup();

      if (group == null) {
        return;
      }

      GroupContainer groupContainer = grouperRequestContainer.getGroupContainer();

      String searchString = request.getParameter("addMemberSubjectSearch");


      boolean searchOk = GrouperUiUtils.searchStringValid(searchString);
      if (!searchOk) {

        guiResponseJs.addAction(GuiScreenAction.newInnerHtml("#addMemberResults",
            TextContainer.retrieveFromRequest().getText().get("groupAddMemberNotEnoughChars")));
        return;
      }

      Set<Subject> subjects = SubjectFinder.findPageInStem(group.getParentStemName(), searchString).
getResults();

      if (GrouperUtil.length(subjects) == 0) {

        guiResponseJs.addAction(GuiScreenAction.newInnerHtml("#addMemberResults",
            TextContainer.retrieveFromRequest().getText().get("groupAddMemberNoSubjectsFound")));
        return;
      }

      Set<GuiSubject> guiSubjects = GuiSubject.convertFromSubjects(subjects, "uiV2.subjectSearchResults", 30);

      groupContainer.setGuiSubjectsAddMember(guiSubjects);

      guiResponseJs.addAction(GuiScreenAction.newInnerHtmlFromJsp("#addMemberResults",
          "/WEB-INF/grouperUi2/group/addMemberResults.jsp"));

    } finally {
      GrouperSession.stopQuietly(grouperSession);
    }
  }
```

sfd

Combobox

To use the new combobox, there are several parts:

Combobox - JSP custom tag

Geneerally you need an idBase (some unique id for combos on that page), a width (via style), and an operation (which can include dynamic params). Note, there are other options like sending other form elements during the filter jax operation

```
<grouper:combobox2 idBase="parentFolderCombo" style="width: 30em"
    filterOperation="../app/UiV2Stem.stemCopyParentFolderFilter?stemId=${grouperRequestContainer.stemContainer.
guiStem.stem.id}"/>
```

Combobox - filter ajax method

There is a helper logic structure to easily do the combobox ajax logic, here is an example:

```
  /**
   * combo filter copy parent folder
   * @param request
   * @param response
   */
  public void stemCopyParentFolderFilter(final HttpServletRequest request, HttpServletResponse response) {

    //run the combo logic
    DojoComboLogic.logic(request, response, new DojoComboQueryLogicBase<Stem>() {

      /**
       *
       */
      @Override
      public Stem lookup(HttpServletRequest request, GrouperSession grouperSession, String query) {
        Subject loggedInSubject = grouperSession.getSubject();
        Stem theStem = new StemFinder().addPrivilege(NamingPrivilege.STEM).assignSubject(loggedInSubject)
            .assignFindByUuidOrName(true).assignScope(query).findStem();
        return theStem;
      }

      /**
       *
       */
      @Override
      public Collection<Stem> search(HttpServletRequest request, GrouperSession grouperSession, String query) {
        Subject loggedInSubject = grouperSession.getSubject();
        QueryOptions queryOptions = QueryOptions.create(null, null, 1, 50);
        return new StemFinder().addPrivilege(NamingPrivilege.STEM).assignScope(query).assignSubject
(loggedInSubject)
            .assignSplitScope(true).assignQueryOptions(queryOptions).findStems();
      }

      /**
       *
       * @param t
       * @return
       */
      @Override
      public String retrieveId(GrouperSession grouperSession, Stem t) {
        return t.getId();
      }

      /**
       *
       */
      @Override
      public String retrieveLabel(GrouperSession grouperSession, Stem t) {
        return t.getDisplayName();
      }

      /**
       *
       */
      @Override
      public String retrieveHtmlLabel(GrouperSession grouperSession, Stem t) {
        //description could be null?
        String label = GrouperUiUtils.escapeHtml(t.getDisplayName(), true);
        String htmlLabel = "<img src=\"../../grouperExternal/public/assets/images/folder.gif\" /> " + label;
        return htmlLabel;
```

```
    }

    /**
     *
     */
    @Override
    public String initialValidationError(HttpServletRequest request, GrouperSession grouperSession) {
      Stem stem = retrieveStemHelper(request, true).getStem();

      if (stem == null) {
        return TextContainer.retrieveFromRequest().getText().get("stemCopyInsufficientPrivileges");
      }
      return null;
    }
  });

}
```

Combobox submit form

Then the form that has the combobox is submitted, process the data. You do this by appending "Name" to the id label. Also, if the user typed something in, but didnt select anything, append "NameDisplay" to the id base.

```
String parentFolderId = request.getParameter("parentFolderComboName");

//just get what they typed in
if (StringUtils.isBlank(parentFolderId)) {
  parentFolderId = request.getParameter("parentFolderComboNameDisplay");
}
```

Combobox control from ajax

If you need to set an ID and label to the combobox from ajax, you can do this: (not sure if this works btw)

```
dijit.byId('parentFolderComboId').set('value', 123);
dijit.byId('parentFolderComboId').set('displayedValue', 'Some label');
```

dfs