

Grouper UI text proposal

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

Text in the Grouper UI can change in the 2.2 redesign.

Currently we use resource bundles, but we could switch to the new config overlays instead of Java resource bundles.

The UI text would be externalized and internationalized.

The config files are similar to the project config files, in that they have configurable overlays. These config files are refreshed every so often, so they can be edited (or patched) on the server without a restart.

For example, the default english/US text config file is on the classpath in: grouperText/grouperUi.text.en.us.base.properties

You should not edit the base text file, you should edit the overlay. In this case the overlay is specified in the config file

```
text.config.hierarchy = classpath:grouperText/grouperUi.text.en.us.base.properties, classpath:grouperText/grouperUi.text.en.us.properties
```

So the file you should edit is: grouperUi.text.en.us.properties. Any text item in the base can be overridden here.

Example of text override

In the base (grouperUi.text.en.us.base.properties), is:

```
# Name of the username
groupCapitalized = Group
```

At Penn, this is called a PennGroup, so in the grouperUi.text.en.us.properties, we could have:

```
# Name of the username
groupCapitalized = PennGroup
```

Referencing text in other text. You can do a quick reference of another text item in a text item like this:

```
# service name with first letter cap
serviceNameCap = Grouper
#####
## Index page: grouperIndex.jsp
#####
# title in browser on index page
indexTitle = $$serviceNameCap$$ index
# subheader in bold if user is opted in
indexSettingsSubheader = $$serviceNameCap$$ settings
```

Dynamic data in text

When text is calculated for a user, all of the text references are resolved for the HTTP request at once. Each time a text item is used, if there is an EL scriptlet, it will be calculated only for that text item call. Here is an example:

```
# Success message when members added to group. The number of members is in ${grouperRequestContainer.addMemberContainer.numberOfUsers}, the uigroup is ${grouperRequestContainer.addMemberContainer.uiGroup}
addMembersSuccess = ${grouperRequestContainer.addMemberContainer.numberOfUsers} were added to ${grouperUtil.escapeXml(grouperRequestContainer.addMemberContainer.uiGroup.name)}
```

Using text in JSP

There are a few calls to use text in JSP. The default one needs no escaping:

```
<div class="formLabel"><b>${textContainer.text['profileEmailLabel']}</b></div>
```

If you want HTML escaped, you can call

```
<a href="../../grouperUnprotectedUi/UiMainUnprotected.logout">${textContainer.textEscapeXml['buttonLogOut']}</a>
```

If you want double quotes escaped for HTML use:

```
<input value="${textContainer.textEscapeDouble['buttonManageSettings']}" class="grouperBlueButton" type="submit" />
```

If you want single quotes escaped for javascript use textEscapeSingle, but most likely in that case you want single and double quotes escaped:

```
onclick="return confirm('${textContainer.textEscapeSingleDouble['adminUntrustConfirm']}');" />
```

Using text in Java

To use text in Java, call the following. Note, if there are variables needed in the text, set those in the request container first.

```
grouperRequestContainer.getAddMemberContainer().setNumberOfUsers(GrouperUtil.length(users));
grouperRequestContainer.getAddMemberContainer().setUiGroup(uiGroup);
String error = TextContainer.retrieveFromRequest().getText().get("addMembersSuccess");
grouperRequestContainer.setError(error);
```