

JEXL script tester

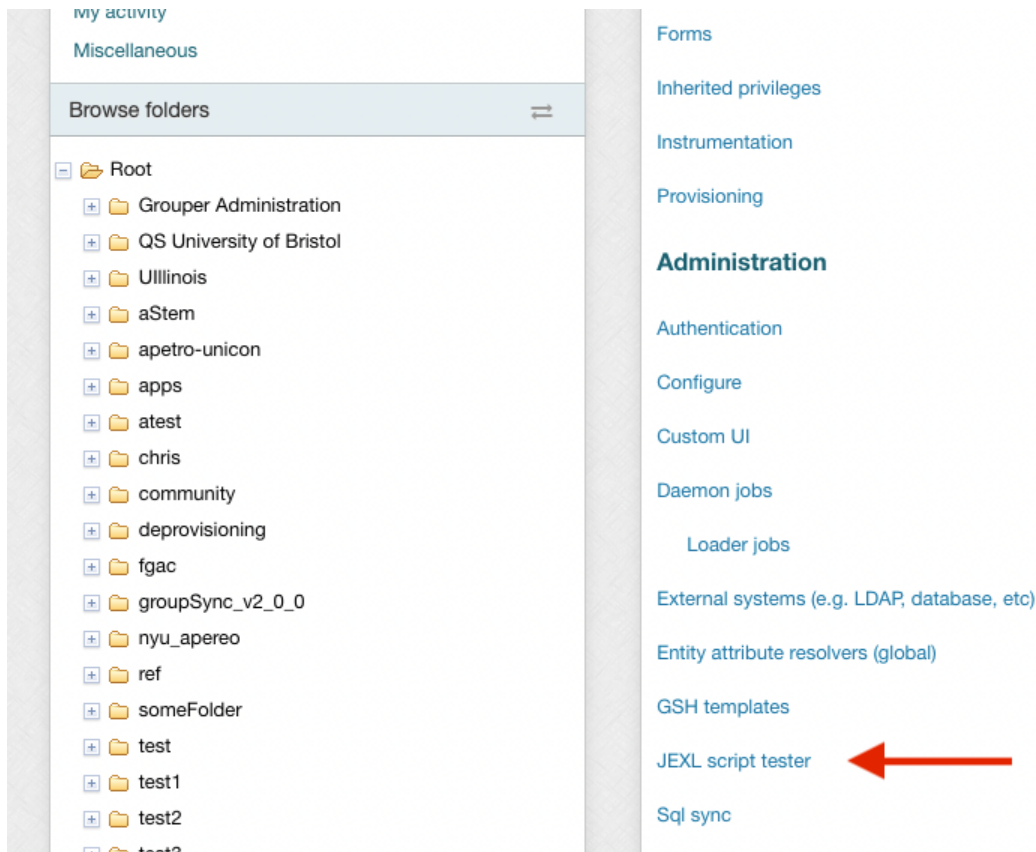
Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

In Grouper v4.0.3+ there is a UI utility to test JEXL scripts.

The point is to test JEXL scripts, and also to show various example in the places where JEXL scripts are used. If you have a good example of a JEXL script please submit it to the Grouper team to add to the screen

Use

Go to Misc -> JEXL script tester

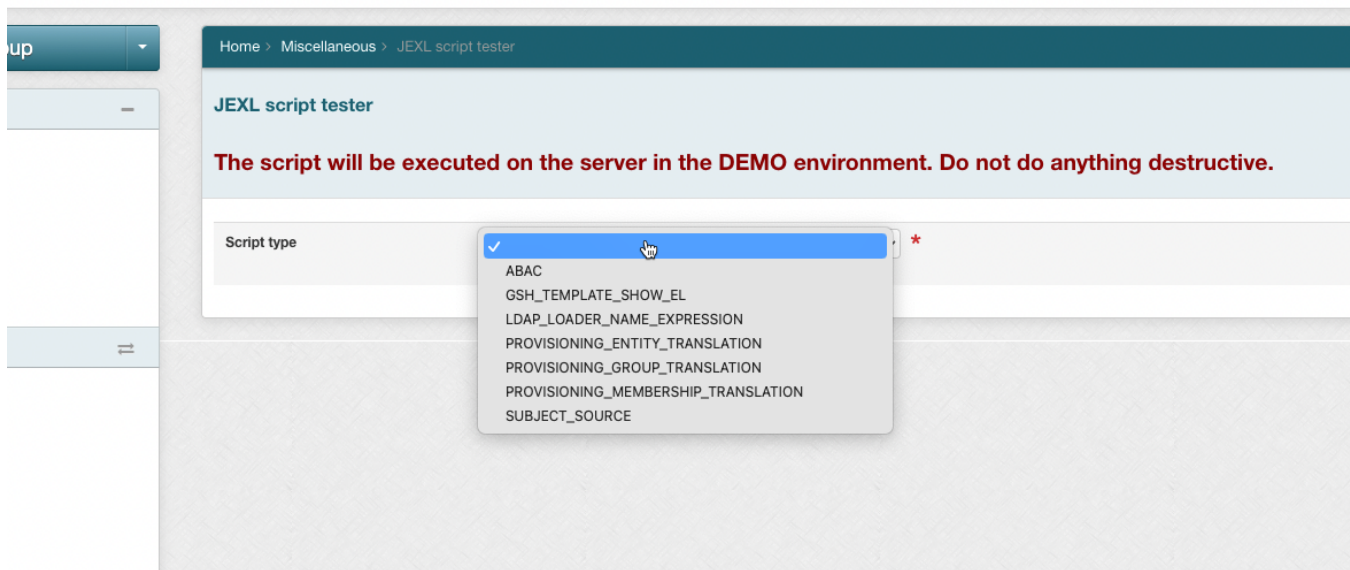


In order to use the JEXL script tester you need to be a Grouper admin and be in the group: etc:jexlScriptTestingGroup

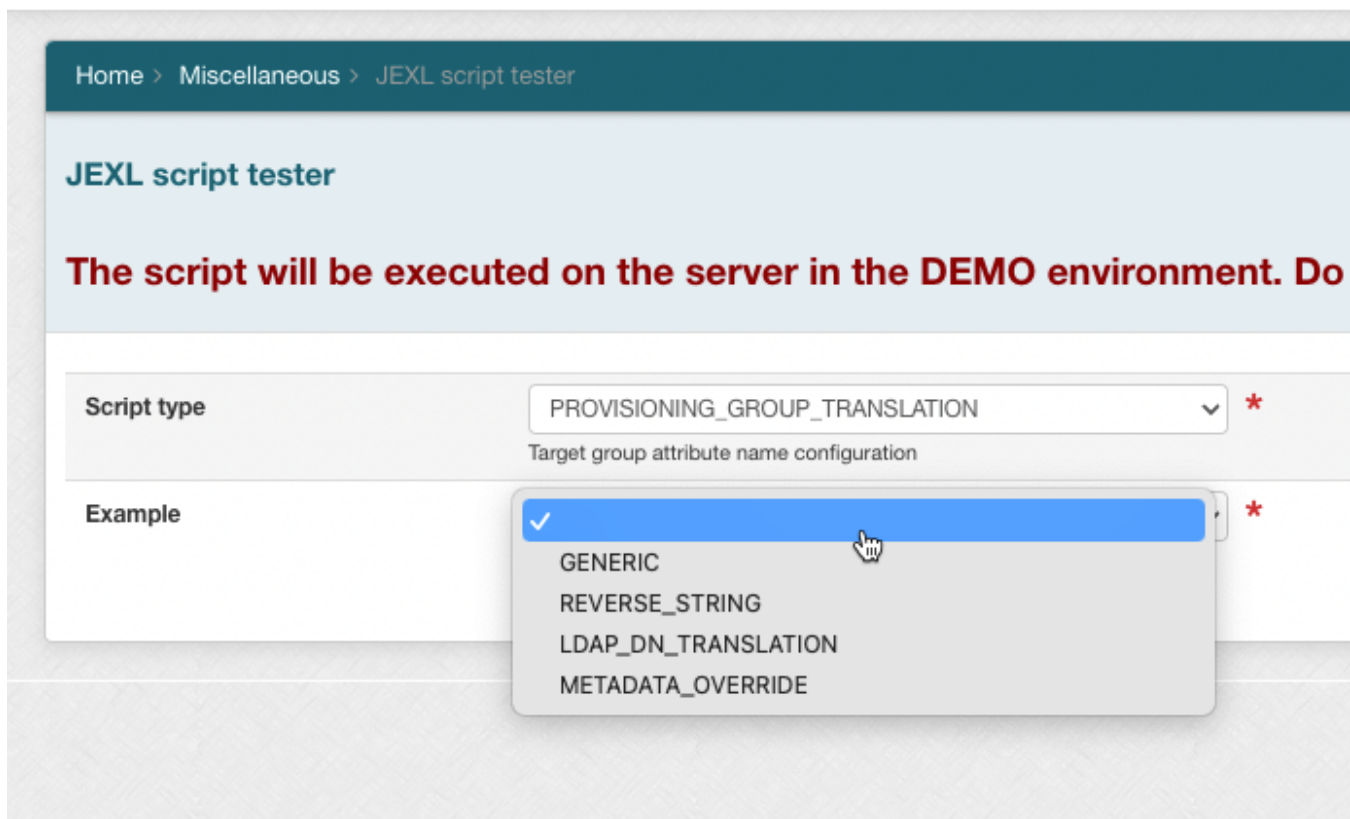
Warning

These JEXL scripts via GSH will run in the UI in the environment where the UI is running. Do not code destructive GSH / JEXL or bad things will happen.

Select the type of script to run



Select the example based on the type of script



Customize 'Available beans'

The 'available beans' section of the screen will pre-populate beans that are in scope for the particular JEXL script. You can customize the data based on what you are testing (in this case, edit the group name)

Available beans GSH script

Null checking JEXL script

```
#{grouperProvisioningGroup.name != null}
```

You could put a JEXL script here to check for nulls safely because JEXL will throw exception if any variables are null e.g. false

Script

```
#{edu.internet2.middleware.grouper.util.GrouperUtil.stringFormatNameReverseReplaceTruncate(grouperProvisioningGroup.name, ".", 64)}
```

Edit the example JEXL script (or submit as is)

Script

```
#{edu.internet2.middleware.grouper.util.GrouperUtil.stringFormatNameReverseReplaceTruncate(grouperProvisioningGroup.name, ".", 64)}
```

Add your script here

See the result

Note the 'type' of data is printed before the string value. In this example the input is:

```
applications:departmentOfArtsAndSciences:commonResources:servicesOfSchool:policyGroups:groupsForActiveDirectory:
historyDepartmentProfessors
```

The method called is: stringFormatNameReverseReplaceTruncate with length of 64 and use dots as separators. You see the result is it reversed with dots and length less than 64.

Add your script here

Submit

Result

java.lang.String: historyDepartmentProfessors.groupsForActiveDirectory.policyGroup

Gsh script executed

Show/Hide gsh script

GSH source

If you want to see the GSH source to run this yourself, it is printed below

Gsh script executed

Show/Hide gsh script

```
import edu.internet2.middleware.grouper.app.jexlTester.*;
import edu.internet2.middleware.grouper.app.provisioning.*;
import edu.internet2.middleware.grouperClient.jdbc.tableSync.GcGrouperSyncGroup;

ProvisioningGroup grouperProvisioningGroup = new ProvisioningGroup();
grouperProvisioningGroup.setName("applications:departmentOfArtsAndSciences:commonResources:servicesOfSchool:policyGroups:groupsForActiveDirectory:historyDepartmentProfessors");
grouperProvisioningGroup.setIdx("abc123");
grouperProvisioningGroup.setIdxIndex(1234567L);

ProvisioningGroup grouperTargetGroup = new ProvisioningGroup();

GcGrouperSyncGroup gcGrouperSyncGroup = new GcGrouperSyncGroup();

ProvisioningGroupWrapper provisioningGroupWrapper = new ProvisioningGroupWrapper();
provisioningGroupWrapper.setGrouperProvisioningGroup(grouperProvisioningGroup);
provisioningGroupWrapper.setGrouperTargetGroup(grouperTargetGroup);
provisioningGroupWrapper.setGcGrouperSyncGroup(gcGrouperSyncGroup);

Map<String, Object> elVariableMap = new HashMap<String, Object>();
elVariableMap.put("grouperProvisioningGroup", grouperProvisioningGroup);
elVariableMap.put("provisioningGroupWrapper", provisioningGroupWrapper);
elVariableMap.put("grouperTargetGroup", grouperTargetGroup);
elVariableMap.put("gcGrouperSyncGroup", gcGrouperSyncGroup);

Object result = null;
String nullCheckingJexlScript = "" + '$' + "{grouperProvisioningGroup.name != null}";
boolean shouldContinue = ScriptType.PROVISIONING_GROUP_TRANSLATION.nullCheckingShouldContinue(elVariableMap, nullCheckingJexlScript);
if (shouldContinue) {
    String jexlScript = "" + '$' + "{edu.internet2.middleware.grouper.util.GrouperUtil.stringFormatNameReverseReplaceTruncate(grouperProvisioningGroup.name, '\\.\\', 64)}";
    result = ScriptType.PROVISIONING_GROUP_TRANSLATION.runJexl(elVariableMap, jexlScript);
    JexlScriptTester.registerOutput(result);
} else {
    JexlScriptTester.registerOutput("Null checking script returned false so the main JEXL script did not execute.");
}
```

Jexl advice

Jexl java bean property names

It is better to use the getter name than the property name. JEXL will throw an exception if there is a null and referring to it as property name.

For example, this will return null

```
ProvisioningGroup provisioningGroup = new ProvisioningGroup();

Map<String, Object> variableMap = new HashMap<String, Object>();
variableMap.put("provisioningGroup", provisioningGroup);

String result = GrouperUtil.substituteExpressionLanguage("${provisioningGroup.getName()}", variableMap,
true, false, false);

System.out.println(result);
```

But this will throw an error

```
ProvisioningGroup provisioningGroup = new ProvisioningGroup();

Map<String, Object> variableMap = new HashMap<String, Object>();
variableMap.put("provisioningGroup", provisioningGroup);

String result = GrouperUtil.substituteExpressionLanguage("${provisioningGroup.name}", variableMap, true,
false, false);

System.out.println(result);
```

See Also

[Grouper ABAC Script Analysis](#)