

Grouper custom template via GSH Internet2 example

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

- [Grouper custom template via GSH Internet2 example delete](#)

COmanage will call a grouper web service to execute a template to create a working group. This is not an exact example but all the necessary features are here so ChrisHu can use these techniques for jira/confluence/etc

template config id: createNewWorkingGroup

group allowed to run this template: ref:workinggroupadmins

Inputs:

Name	Type	Validation	Default	Required	Description
gsh_input_workingGroupExtension	String	alphanumeric and dash		required	will be used in the folder and group system names
gsh_input_workingGroupDisplayExtension	String	no colon		optional	can be used as display extension, or default to extension
gsh_input_workingGroupDescription	String	none		optional	can be used as description for folders/groups
gsh_input_isSympa	boolean	NA	false	optional	if its a sympy enabled working group
gsh_input_sympaDomain	String	internet2 incommon		required if gsh_input_isSympa	(dropdown) which domain the list is in
gsh_input_isSympaModerated	boolean	NA	false	optional	if sympy list is moderated add attribute to sympy working group folder
gsh_input_isOptin	boolean		false	optional	if the users list can be optin from all
gsh_input_attestationDays	int	between 30 and 365	365	optional	days for attestation
gsh_input_isConfluence	boolean		false		if confluence groups should be built out
gsh_input_isJira	boolean		false		if Jira groups should be built out

GSH actions (in a transaction):

Step	Run if condition	Description	Message
Check if working group folder exists	always	Return an error and fail execution with descriptive message	Error: working group extension 'abc' already exists!
Check gsh_input_sympaDomain	if gsh_input_isSympa	Return an error and fail execution with descriptive message	Error: Sympa domain is required if provisioning to sympy
Create working group folder	always	Create working group folder: ref:incommon-collab: <gsh_input_workingGroupExtension> User friendly name and description if applicable for this folder and subsequent objects	Folder created: ref:incommon-collab: <gsh_input_workingGroupExtension>
Create users group	always	Create users group: ref:incommon-collab: <gsh_input_workingGroupExtension>:users	Group created: ref:incommon-collab: <gsh_input_workingGroupExtension>:users
Create admins group	always	Create admins group: ref:incommon-collab: <gsh_input_workingGroupExtension>:admins	Group created: ref:incommon-collab: <gsh_input_workingGroupExtension>:admins
Resolve admin subject	if not blank gsh_input_initialAdminSubjectId	If not resolvable, give warning	Warning: admin subject not resolvable '<subject_id>'
Add admin to group	if certain source? gsh_builtin_subject	Add admin subject to admins group	Print message about user being added
Assign privs for admins on folder	always	Assign inherited group admin privs on working group folder for admin group	Assigned group admin privileges to ref:incommon-collab: <gsh_input_workingGroupExtension>:admins inherited from folder: ref:incommon-collab: <gsh_input_workingGroupExtension>

Assign attestation at working group folder level for admins	always	Add attestation at working group folder for admins to attest	Added attestation for 90 days on groups in folder: ref:incommon-collab: <gsh_input_workingGroupExtension>
Create sympa folder	if gsh_input_isSympa	Create folder app:sympa:<domain>: <gsh_input_workingGroupExtension>	Folder created: app:sympa:<domain>: <gsh_input_workingGroupExtension>
Create owners sympa group	if gsh_input_isSympa	Create owners group: create group app:sympa:<domain>: <gsh_input_workingGroupExtension>:owners	Group created: app:sympa:<domain>: <gsh_input_workingGroupExtension>:owners
Create subscribers sympa group	if gsh_input_isSympa	Create subscribers group: create group app:sympa:<domain>: <gsh_input_workingGroupExtension>:subscribers	Group created: app:sympa:<domain>: <gsh_input_workingGroupExtension>:subscribers
Add admin group to owners group	if gsh_input_isSympa	Add ref:incommon-collab: <gsh_input_workingGroupExtension>:admins to be member of app:sympa:<domain>: <gsh_input_workingGroupExtension>:owners	Added member ref:incommon-collab: <gsh_input_workingGroupExtension>:admins to group app:sympa:<domain>: <gsh_input_workingGroupExtension>:owners
Add users group to subscribers group	if gsh_input_isSympa	Add ref:incommon-collab: <gsh_input_workingGroupExtension>:users to be member of app:sympa:<domain>: <gsh_input_workingGroupExtension>:subscribers	Added member ref:incommon-collab: <gsh_input_workingGroupExtension>:users to group app:sympa:<domain>: <gsh_input_workingGroupExtension>:subscribers
Assign privs for admins on sympa folder	if gsh_input_isSympa	Assign inherited group admin privs on sympa folder for admin group	Assigned group admin privileges to ref:incommon-collab: <gsh_input_workingGroupExtension>:admins inherited from folder: app:sympa:<domain>: <gsh_input_workingGroupExtension>
Add moderated marker attribute to sympa folder	if gsh_input_isSympa and gsh_input_isSympaModerated	Assign a sympa attribute to the sympa folder for midpoint	Assigned attribute app:sympa:attribute:moderated to folder app:sympa:<domain>: <gsh_input_workingGroupExtension>
Assign attestation at sympa folder level for admins	if gsh_input_isSympa	Add attestation at sympa folder for admins to attest	Added attestation for 90 days on groups in folder: app:sympa:<domain>: <gsh_input_workingGroupExtension>

Common settings

Setting	Value	Description
actAs (WS)	some user	audits will be correct if
runAs	GrouperSystem	template runs as privileged user
canRun	a:b:c	make a group and put the COnfigure credential in there and all users who will be running template
transactional	true	if something fails, roll the whole thing back
individual audits	true	each individual action should be audited
runFromFolder	ref:incommon-collab	you will see this option in the menu when on that folder

Home > Miscellaneous > GSH templates

GSH templates

Actions ▾

Config id	Enabled	Actions
createNewWorkingGroup	Enabled	Actions ▾

Config screen

GSH templates

Actions ▾

Config id	createNewWorkingGroup		
Enabled	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (True) <input type="radio"/> True <input type="radio"/> False if this template is enabled. Default value is 'true'.	
Show on groups	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (False) <input type="radio"/> True <input type="radio"/> False if this template option is available on groups. Default value is 'false'.	
Show on folders	<input type="checkbox"/> EL?	<input type="radio"/> Default value (False) <input checked="" type="radio"/> True <input type="radio"/> False if this template option is available on folders. Default value is 'false'.	
Folder show type	<input type="checkbox"/> EL?	<div>certainFolder ▾ *</div> <div>Where should this template be available</div>	
Folder uuid to show	<input type="checkbox"/> EL?	<div>228348d9741748b791e243342953ee3f *</div> <div>UUID of folder to show this template</div>	
Folder show on descendants	<input type="checkbox"/> EL?	<div>certainFolder ▾ *</div> <div>certainFolder: just show on one folder. oneChildLevel: only show on child level under folder. certainFolderAndOneChildLevel: show the folder and one level of children. descendants: show all folders under the folder. certainFolderAndDescendants: show folder and all descendants.</div>	
Security run type	<input type="checkbox"/> EL?	<div>specifiedGroup ▾ *</div> <div>Who can run this template. Only GrouperSystem / wheel group, or specify a group</div>	
Group uuid	<input type="checkbox"/> EL?	<div>84c0409d43254c7b81f95bab8f5fc9f4 *</div> <div>UUID of group that can run this template</div>	
Run as type	<input type="checkbox"/> EL?	<div>GrouperSystem ▾ *</div> <div>Select the type of user to run as. "currentUser" means run as the user using the UI. "GrouperSystem" means run as root user, "specifiedSubject" means you can pick a subject to run as (not common).</div>	
Externalized text?	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (False) <input type="radio"/> True <input type="radio"/> False Select 'True' if you would like to use externalized text for template name, description, and input labels & descriptions. Default value is 'false'.	
Template name	<input type="checkbox"/> EL?	<div>Create new working group *</div> <div>Template name</div>	
Template description	<input type="checkbox"/> EL?	<div>Add a new working group including collaboration tools</div> <div>Template description</div>	
Lightweight GSH	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (False) <input type="radio"/> True <input type="radio"/> False If you are managing your own imports and don't want built-ins select true. This reduces the GSH execution time by a few seconds. Default value is 'false'.	
Run gsh script in transaction	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (True) <input type="radio"/> True <input type="radio"/> False Run gsh script in a database transaction so that if any statement fails, then all the database changes are rolled back. Default value is 'true'.	
GSH script	<input type="checkbox"/> EL?	<div>String workingGroupFolderName = "sandbox:ref.incommon- collab:" + gsh_input_workingGroupExtension; // default value for display extension is just the extension This is the GSH template to run</div>	
Number of inputs	<input type="checkbox"/> EL?	<div>7 ▾</div>	

Number of inputs (form elements on ui or in ws). Default value is '0'.

Input item - 1

Input item

Input name - 1 ☐ EL? *

Name of the ui form element, ws param, template variable

Label - 1 ☐ EL?

Label on the UI for this input

Description - 1 ☐ EL?

Description on the UI for this input

Input type - 1 ☐ EL?

Type of the data. Default value is 'string'.

Form element type - 1 ☐ EL?

Form element type on UI. Default value is 'textfield'.

Form element index - 1 ☐ EL?

Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.

Validation type - 1 ☐ EL? *

Type of validation on the input

Builtin validation - 1 ☐ EL? *

Type of builtin validation on the input

input.0.validationMessage ☐ EL?

validation message text

Required - 1 ☐ EL? ☐ Default value (False) ☒ True ☐ False

If this input is required for template to run. Default value is 'false'.

Jexl for showEl - 1 ☐ EL?

Jexl for if this field should show, note all inputs are available to use as variables

Trim whitespace - 1 ☐ EL? ☒ Default value (True) ☐ True ☐ False

If whitespace should be trimmed from the value for this input. Default value is 'true'.

Input item - 1

Input item

Input name - 2 ☐ EL? *

Name of the ui form element, ws param, template variable

Label - 2 ☐ EL?

Label on the UI for this input

Description - 2 ☐ EL?

Description on the UI for this input

Input type - 2 ☐ EL?
Type of the data. Default value is 'string'.

Form element type - 2 ☐ EL?
Form element type on UI. Default value is 'textfield'.

Form element index - 2 ☐ EL?

Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.

Validation type - 2 ☐ EL? *

Type of validation on the input

Validation regex - 2 ☐ EL? *

Regex to check the input and if it doesnt match then fail. For example, this is alphanumeric or underscore length between 1 and 50

input.1.validationMessage ☐ EL?

validation message text

Required - 2 ☐ EL? ☒ Default value (False) ☐ True ☐ False

If this input is required for template to run. Default value is 'false'.

Default value - 2 ☐ EL?

default value for the input if none is provided

Jexl for showEI - 2 ☐ EL?

Jexl for if this field should show, note all inputs are available to use as variables

Trim whitespace - 2 ☐ EL? ☒ Default value (True) ☐ True ☐ False

If whitespace should be trimmed from the value for this input. Default value is 'true'.

Input item - 1

Input item

Input name - 3 ☐ EL? *

Name of the ui form element, ws param, template variable

Label - 3 ☐ EL?

Label on the UI for this input

Description - 3 ☐ EL?

Description on the UI for this input

Input type - 3 ☐ EL?
Type of the data. Default value is 'string'.

Form element type - 3 ☐ EL?

Form element type on UI. Default value is 'textfield'.

Form element index - 3

☐ EL?

Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.

Validation type - 3

☐ EL?



Type of validation on the input

input.2.validationMessage

☐ EL?

validation message text

Required - 3

☐ EL?

☒ Default value (False) ☐ True ☐ False

If this input is required for template to run. Default value is 'false'.

Default value - 3

☐ EL?

default value for the input if none is provided

Jexl for showEI - 3

☐ EL?

Jexl for if this field should show, note all inputs are available to use as variables

Trim whitespace - 3

☐ EL?

☒ Default value (True) ☐ True ☐ False

If whitespace should be trimmed from the value for this input. Default value is 'true'.

Input item - 1

Input item

Input name - 4

☐ EL?



Name of the ui form element, ws param, template variable

Label - 4

☐ EL?

Label on the UI for this input

Description - 4

☐ EL?

Description on the UI for this input

Input type - 4

☐ EL?



Type of the data. Default value is 'string'.

Form element type - 4

☐ EL?



Form element type on UI. Default value is 'textfield'.

Form element index - 4

☐ EL?

Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.

Form element index - 4	<input type="checkbox"/> EL?	<input type="text"/>	Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.
Validation type - 4	<input type="checkbox"/> EL?	<input type="text" value="none"/> <input type="button" value="v"/> *	Type of validation on the input
input.3.validationMessage	<input type="checkbox"/> EL?	<input type="text"/>	validation message text
Required - 4	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (False) <input type="radio"/> True <input type="radio"/> False	If this input is required for template to run. Default value is 'false'.
Default value - 4	<input type="checkbox"/> EL?	<input type="text" value="false"/>	default value for the input if none is provided
Jexl for showEI - 4	<input type="checkbox"/> EL?	<input type="text"/>	Jexl for if this field should show, note all inputs are available to use as variables
Trim whitespace - 4	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (True) <input type="radio"/> True <input type="radio"/> False	If whitespace should be trimmed from the value for this input. Default value is 'true'.
Input item - 1 Input item			
Input name - 5	<input type="checkbox"/> EL?	<input type="text" value="gsh_input_sympaDomain"/> *	Name of the ui form element, ws param, template variable
Label - 5	<input type="checkbox"/> EL?	<input type="text" value="Sympa domain"/>	Label on the UI for this input
Description - 5	<input type="checkbox"/> EL?	<input type="text" value="incommon or internet2 domain on email list"/>	Description on the UI for this input
Input type - 5	<input type="checkbox"/> EL?	<input type="text"/> <input type="button" value="v"/>	Type of the data. Default value is 'string'.
Form element type - 5	<input type="checkbox"/> EL?	<input type="text" value="dropdown"/> <input type="button" value="v"/>	Form element type on UI. Default value is 'textfield'.
Form element index - 5	<input type="checkbox"/> EL?	<input type="text"/>	Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.
Validation type - 5	<input type="checkbox"/> EL?	<input type="text" value="regex"/> <input type="button" value="v"/> *	Type of validation on the input
Validation regex - 5	<input type="checkbox"/> EL?	<input type="text" value="^(incommon internet2)\$"/> *	Regex to check the input and if it doesnt match then fail. For example, this is alphanumeric or underscore length between 1 and 50
input.4.validationMessage	<input type="checkbox"/> EL?	<input type="text"/>	validation message text
Required - 5	<input type="checkbox"/> EL?	<input type="radio"/> Default value (False) <input checked="" type="radio"/> True <input type="radio"/> False	If this input is required for template to run. Default value is 'false'.
Jexl for showEI - 5	<input type="checkbox"/> EL?	<input type="text" value="{gsh_input_isSympa}"/>	Jexl for if this field should show, note all inputs are available to use as variables
Trim whitespace - 5	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (True) <input type="radio"/> True <input type="radio"/> False	

If whitespace should be trimmed from the value for this input. Default value is 'true'.

Input item - 1

Input item

Input name - 6

☐ EL?

gsh_input_isSympaModerated

*

Name of the ui form element, ws param, template variable

Label - 6

☐ EL?

Moderated?

Label on the UI for this input

Description - 6

☐ EL?

Is this sympa email list "moderated"? i.e. does someone need to

Description on the UI for this input

Input type - 6

☐ EL?

boolean



Type of the data. Default value is 'string'.

Form element type - 6

☐ EL?



Form element type on UI. Default value is 'textfield'.

Form element index - 6

☐ EL?

Form elements will show on the form in order of "index". Default Value is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.

Validation type - 6

☐ EL?

none



*

Type of validation on the input

input.5.validationMessage

☐ EL?

validation message text

Required - 6

☐ EL?

☒ Default value (False) ☐ True ☐ False

If this input is required for template to run. Default value is 'false'.

Default value - 6

☐ EL?

false

default value for the input if none is provided

Jexl for showEI - 6

☐ EL?

\${gsh_input_isSympa}

Jexl for if this field should show, note all inputs are available to use as variables

Trim whitespace - 6

☐ EL?

☒ Default value (True) ☐ True ☐ False

If whitespace should be trimmed from the value for this input. Default value is 'true'.

Input item - 1

Input item

Input name - 7	<input type="checkbox"/> EL?	<input type="text" value="gsh_input_initialAdminSubjectId"/> *
Name of the ui form element, ws param, template variable		
Label - 7	<input type="checkbox"/> EL?	<input type="text" value="Initial admin subject ID"/>
Label on the UI for this input		
Description - 7	<input type="checkbox"/> EL?	<input type="text" value="Enter the initial admin subject ID to add to admins group"/>
Description on the UI for this input		
Input type - 7	<input type="checkbox"/> EL?	<input type="text" value="string"/> ▼
Type of the data. Default value is 'string'.		
Form element type - 7	<input type="checkbox"/> EL?	<input type="text" value="textfield"/> ▼
Form element type on UI. Default value is 'textfield'.		
Form element index - 7	<input type="checkbox"/> EL?	<input type="text" value="0"/>
Form elements will show on the form in order of "index". DefaultValue is 0. If two elements have the same index then they will show in order of configuration. Default value is '0'.		
Validation type - 7	<input type="checkbox"/> EL?	<input type="text" value="regex"/> ▼ *
Type of validation on the input		
Validation regex - 7	<input type="checkbox"/> EL?	<input type="text" value="^[^s]+\$"/> *
Regex to check the input and if it doesnt match then fail. For example, this is alphanumeric or underscore length between 1 and 50		
input.6.validationMessage	<input type="checkbox"/> EL?	<input type="text" value="Do not use whitespace in admin subject ID"/>
validation message text		
Required - 7	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (False) <input type="radio"/> True <input type="radio"/> False
If this input is required for template to run. Default value is 'false'.		
Default value - 7	<input type="checkbox"/> EL?	<input type="text" value=""/>
default value for the input if none is provided		
Jexl for showEI - 7	<input type="checkbox"/> EL?	<input type="text" value=""/>
Jexl for if this field should show, note all inputs are available to use as variables		
Trim whitespace - 7	<input type="checkbox"/> EL?	<input checked="" type="radio"/> Default value (True) <input type="radio"/> True <input type="radio"/> False
If whitespace should be trimmed from the value for this input. Default value is 'true'.		
Test settings		
Test settings		
Number of tests	<input type="checkbox"/> EL?	<input type="text" value="0"/> ▼
Number of tests to make sure this template functions. Default value is '0'.		

Configuration

```

grouperGshTemplate.createNewWorkingGroup.folderShowOnDescendants = certainFolder
grouperGshTemplate.createNewWorkingGroup.folderShowType = certainFolder
grouperGshTemplate.createNewWorkingGroup.folderUuidToShow = 228348d9741748b791e243342953ee3f
grouperGshTemplate.createNewWorkingGroup.groupUuidCanRun = 84c0409d43254c7b81f95bab8f5fc9f4
grouperGshTemplate.createNewWorkingGroup.gshTemplate = String workingGroupFolderName = "ref:incommon-collab:" +
gsh_input_workingGroupExtension;\
\
// default value for display extension is just the extension\
String displayExtension = GrouperUtil.defaultIfBlank(gsh_input_workingGroupDisplayExtension,
gsh_input_workingGroupExtension);\
\
// we dont want the word "null" printed out if this is null, so convert to empty string if this is null\
gsh_input_workingGroupDescription = GrouperUtil.defaultString(gsh_input_workingGroupDescription);\
...

grouperGshTemplate.createNewWorkingGroup.input.0.description = Working group extension is the Grouper system
name of the folders
grouperGshTemplate.createNewWorkingGroup.input.0.label = Working group extension
grouperGshTemplate.createNewWorkingGroup.input.0.name = gsh_input_workingGroupExtension
grouperGshTemplate.createNewWorkingGroup.input.0.required = true
grouperGshTemplate.createNewWorkingGroup.input.0.validationBuiltin = alphaNumericDash
grouperGshTemplate.createNewWorkingGroup.input.0.validationMessage = Only enter alphanumeric or dash
grouperGshTemplate.createNewWorkingGroup.input.0.validationType = builtin
grouperGshTemplate.createNewWorkingGroup.input.1.description = Optionally enter a display extension
grouperGshTemplate.createNewWorkingGroup.input.1.label = Display extension
grouperGshTemplate.createNewWorkingGroup.input.1.name = gsh_input_workingGroupDisplayExtension
grouperGshTemplate.createNewWorkingGroup.input.1.validationMessage = Do not enter colons in display extension
grouperGshTemplate.createNewWorkingGroup.input.1.validationRegex = ^[^\:]+$
grouperGshTemplate.createNewWorkingGroup.input.1.validationType = regex
grouperGshTemplate.createNewWorkingGroup.input.2.description = Give context about what this working group is,
will be put in Grouper descriptions
grouperGshTemplate.createNewWorkingGroup.input.2.label = Description
grouperGshTemplate.createNewWorkingGroup.input.2.name = gsh_input_workingGroupDescription
grouperGshTemplate.createNewWorkingGroup.input.2.validationType = none
grouperGshTemplate.createNewWorkingGroup.input.3.defaultValue = false
grouperGshTemplate.createNewWorkingGroup.input.3.description = If this working group has a sympa mailing list
grouperGshTemplate.createNewWorkingGroup.input.3.label = Is sympa mail
grouperGshTemplate.createNewWorkingGroup.input.3.name = gsh_input_isSympa
grouperGshTemplate.createNewWorkingGroup.input.3.type = boolean
grouperGshTemplate.createNewWorkingGroup.input.3.validationType = none
grouperGshTemplate.createNewWorkingGroup.input.4.description = incommon or internet2 domain on email list
grouperGshTemplate.createNewWorkingGroup.input.4.formElementType = dropdown
grouperGshTemplate.createNewWorkingGroup.input.4.label = Sympa domain
grouperGshTemplate.createNewWorkingGroup.input.4.name = gsh_input_sympaDomain
grouperGshTemplate.createNewWorkingGroup.input.4.required = true
grouperGshTemplate.createNewWorkingGroup.input.4.showEl = ${gsh_input_isSympa}
grouperGshTemplate.createNewWorkingGroup.input.4.validationRegex = ^(incommon|internet2)$
grouperGshTemplate.createNewWorkingGroup.input.4.validationType = regex
grouperGshTemplate.createNewWorkingGroup.input.5.defaultValue = false
grouperGshTemplate.createNewWorkingGroup.input.5.description = Is this sympa email list "moderated"? i.e. does
someone need to approve emails?
grouperGshTemplate.createNewWorkingGroup.input.5.label = Moderated?
grouperGshTemplate.createNewWorkingGroup.input.5.name = gsh_input_isSympaModerated
grouperGshTemplate.createNewWorkingGroup.input.5.showEl = ${gsh_input_isSympa}
grouperGshTemplate.createNewWorkingGroup.input.5.type = boolean
grouperGshTemplate.createNewWorkingGroup.input.5.validationType = none
grouperGshTemplate.createNewWorkingGroup.numberOfInputs = 6
grouperGshTemplate.createNewWorkingGroup.runAsType = GrouperSystem
grouperGshTemplate.createNewWorkingGroup.securityRunType = specifiedGroup
grouperGshTemplate.createNewWorkingGroup.showOnFolders = true
grouperGshTemplate.createNewWorkingGroup.templateDescription = Add a new working group including collaboration
tools
grouperGshTemplate.createNewWorkingGroup.templateName = Create new working group

```

GSH template script

```
import edu.internet2.middleware.grouper.app.attestation.*;
import edu.internet2.middleware.grouper.attr.assign.*;
import edu.internet2.middleware.grouper.privs.*;

String workingGroupFolderName = "ref:incommon-collab:" + gsh_input_workingGroupExtension;

// default value for display extension is just the extension
String displayExtension = GrouperUtil.defaultIfBlank(gsh_input_workingGroupDisplayExtension,
gsh_input_workingGroupExtension);

// we dont want the word "null" printed out if this is null, so convert to empty string if this is null
gsh_input_workingGroupDescription = GrouperUtil.defaultString(gsh_input_workingGroupDescription);

// validate first
// Check if working group folder exists
Stem workingGroupFolder = StemFinder.findByName(gsh_builtin_grouperSession, workingGroupFolderName, false);
if (workingGroupFolder != null) {
    gsh_builtin_gshTemplateOutput.addValidationLine("gsh_input_workingGroupExtension","Error: working group
extension '" + gsh_input_workingGroupExtension + "' already exists!");
}

// Check gsh_input_sympaDomain if gsh_input_isSympa
if (gsh_input_isSympa && GrouperUtil.isBlank(gsh_input_sympaDomain)) {
    gsh_builtin_gshTemplateOutput.addValidationLine("gsh_input_sympaDomain","Error: Sympa domain is required if
provisioning to Sympa");
}

String sympafolderName = null;

if (gsh_input_isSympa) {
    sympafolderName = "app:sympa:" + gsh_input_sympaDomain + ":" + gsh_input_workingGroupExtension;
    Stem sympafolder = StemFinder.findByName(gsh_builtin_grouperSession, sympafolderName, false);
    if (sympafolder != null) {
        gsh_builtin_gshTemplateOutput.addValidationLine("gsh_input_workingGroupExtension", "Error: sympafolder '"
+ sympafolderName + "' already exists!");
    }
}

String confluenceFolderName = null;

if (gsh_input_isConfluence) {
    confluenceFolderName = "app:confluence:" + gsh_input_workingGroupExtension;
    Stem confluenceFolder = StemFinder.findByName(gsh_builtin_grouperSession, confluenceFolderName, false);
    if (confluenceFolder != null) {
        gsh_builtin_gshTemplateOutput.addValidationLine("gsh_input_workingGroupExtension","Error: confluence folder
'" + confluenceFolderName + "' already exists!");
    }
}

String jiraFolderName = null;

if (gsh_input_isJira) {
    jiraFolderName = "app:jira:" + gsh_input_workingGroupExtension;
    Stem jiraFolder = StemFinder.findByName(gsh_builtin_grouperSession, jiraFolderName, false);
    if (jiraFolder != null) {
        gsh_builtin_gshTemplateOutput.addValidationLine("gsh_input_workingGroupExtension","Error: jira folder '"
+ jiraFolderName + "' already exists!");
    }
}

// Do not proceed if there is an error
if (GrouperUtil.length(gsh_builtin_gshTemplateOutput.getValidationLines()) > 0) {
    gsh_builtin_gshTemplateOutput.assignIsError(true);
    GrouperUtil.gshReturn();
}

// Create working group folder
Stem workingGroupFolder = new StemSave().assignName(workingGroupFolderName).assignDisplayExtension
```

```

(displayExtension).assignDescription("Folder holds working group roles. " + gsh_input_workingGroupDescription).
save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Folder created: " + workingGroupFolderName);

// Create users group
Group usersGroup = new GroupSave().assignName(workingGroupFolderName + ":users").assignDisplayExtension
(displayExtension + " users").assignDescription("Users role means members of the working group with access to
collaboration tools. " + gsh_input_workingGroupDescription).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + usersGroup.getName());

if (gsh_input_isOptin) {
    usersGroup.grantPriv(SubjectFinder.findAllSubject(), AccessPrivilege.OPTIN, false);
    gsh_builtIn_gshTemplateOutput.addOutputLine("Granted optin to all for: " + usersGroup.getName());

    usersGroup.grantPriv(SubjectFinder.findAllSubject(), AccessPrivilege.OPTOUT, false);
    gsh_builtIn_gshTemplateOutput.addOutputLine("Granted optout to all for: " + usersGroup.getName());
}

// Create admins group
Group adminsGroup = new GroupSave().assignName(workingGroupFolderName + ":admins").assignDisplayExtension
(displayExtension + " admins").assignDescription("Admins role means can manage / attest the working group. " +
gsh_input_workingGroupDescription).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + adminsGroup.getName());

// If the initial admin is passed in
if (gsh_builtIn_subject != null && GrouperUtil.equals("ldap", gsh_builtIn_subject.getSourceId())) {

    // add the initial admin to be in the admins group
    new MembershipSave().assignGroup(adminsGroup).assignSubject(gsh_builtIn_subject).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Added admin: " + gsh_builtIn_subject.getId() + " to group: " +
adminsGroup.getName());
} else {
    // if not resolvable thats just a warning
    gsh_builtIn_gshTemplateOutput.addOutputLine("info","Warning: admin subject not resolvable or in wrong source
" + gsh_builtIn_subjectId + "");
}

// Assign privs for admins on folder
new PrivilegeGroupInheritanceSave().assignStem(workingGroupFolder).addPrivilegeName("admin").assignSubject
(adminsGroup.toSubject()).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned group admin privileges to: " + adminsGroup.getName() + "
inherited from folder: " + workingGroupFolder.getName());

// Assign attestation on working group folder
new AttestationStemSave().assignStemName(workingGroupFolderName).assignDaysUntilRecertify
(gsh_input_attestationDays).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned attestation to folder: " + workingGroupFolderName);

// sympa folder
if (gsh_input_isSympa) {
    Stem sympafolder = new StemSave().assignName(sympaFolderName).assignDisplayExtension(displayExtension + "
sympa").assignDescription("Folder email lists for working group. " + gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Folder created: " + sympafolder.getName());

    // Create admins group
    Group ownersGroup = new GroupSave().assignName(sympaFolderName + ":owners").assignDisplayExtension
(displayExtension + " sympafolder owners").assignDescription("Owners list manages the email list for the working
group. " + gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + ownersGroup.getName());

    // Create subscribers group
    Group subscribersGroup = new GroupSave().assignName(sympaFolderName + ":subscribers").assignDisplayExtension
(displayExtension + " sympafolder subscribers").assignDescription("Subscribers list receives working group emails. "
+ gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + subscribersGroup.getName());

    // Add admin group to owners group
    new MembershipSave().assignGroup(ownersGroup).assignSubject(adminsGroup.toSubject()).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Added member: " + adminsGroup.getName() + " to group: " +
ownersGroup.getName());
}

```

```

// Add users group to subscribers group
new MembershipSave().assignGroup(subscribersGroup).assignSubject(usersGroup.toSubject()).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Added member: " + usersGroup.getName() + " to group: " +
subscribersGroup.getName());

// Assign privs for admins on folder
new PrivilegeGroupInheritanceSave().assignStem(sympaFolder).addPrivilegeName("admin").assignSubject
(adminsGroup.toSubject()).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned group admin privileges to: " + adminsGroup.getName() +
" inherited from folder: " + sympaFolder.getName());

// Add moderated marker attribute to sympa folder if gsh_input_isSympa and gsh_input_isSympaModerated
if (gsh_input_isSympaModerated ) {

    // Assign a sympa attribute to the sympa folder for midpoint
    new AttributeAssignToStemSave().assignStem(sympaFolder).assignNameOrAttributeDefName("app:sympa:attribute:
moderated").save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned attribute: app:sympa:attribute:moderated to: " +
sympaFolder.getName());
}

// Assign attestation at sympa
new AttestationStemSave().assignStem(sympaFolder).assignDaysUntilRecertify(gsh_input_attestationDays).save();
gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned attestation to folder: " + sympaFolder.getName());
}

if (gsh_input_isConfluence) {
    Stem confluenceFolder = new StemSave().assignName(confluenceFolderName).assignDisplayExtension
(displayExtension + " confluence").assignDescription("Confluence groups for working group. " +
gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Folder created: " + confluenceFolder.getName());

    // Create admins group
    Group confluenceAdminsGroup = new GroupSave().assignName(confluenceFolderName + ":admins").
assignDisplayExtension(displayExtension + " confluence admins").assignDescription("Admins of confluence space
for working group. " + gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + confluenceAdminsGroup.getName());

    // Create users group
    Group confluenceUsersGroup = new GroupSave().assignName(confluenceFolderName + ":users").
assignDisplayExtension(displayExtension + " confluence users").assignDescription("Users of confluence space for
working group. " + gsh_input_workingGroupDescription).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Group created: " + confluenceUsersGroup.getName());

    // Add admin group to owners group
    new MembershipSave().assignGroup(confluenceAdminsGroup).assignSubject(adminsGroup.toSubject()).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Added member: " + adminsGroup.getName() + " to group: " +
confluenceAdminsGroup.getName());

    // Add users group to subscribers group
    new MembershipSave().assignGroup(confluenceUsersGroup).assignSubject(usersGroup.toSubject()).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Added member: " + usersGroup.getName() + " to group: " +
confluenceUsersGroup.getName());

    // Assign privs for admins on folder
    new PrivilegeGroupInheritanceSave().assignStem(confluenceFolder).addPrivilegeName("admin").assignSubject
(confluenceAdminsGroup.toSubject()).save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned group admin privileges to: " + confluenceAdminsGroup.
getName() + " inherited from folder: " + confluenceFolder.getName());

    // Assign attestation at confluence
    new AttestationStemSave().assignStem(confluenceFolder).assignDaysUntilRecertify(gsh_input_attestationDays).
save();
    gsh_builtIn_gshTemplateOutput.addOutputLine("Assigned attestation to folder: " + confluenceFolder.getName());
}

// jira folder
if (gsh_input_isJira) {

```

```

    Stem jiraFolder = new StemSave().assignName(jiraFolderName).assignDisplayExtension(displayExtension + "
jira").assignDescription("Admins of jira project for working group. " + gsh_input_workingGroupDescription).
save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Folder created: " + jiraFolder.getName());

    // Create admins group
    Group jiraAdminsGroup = new GroupSave().assignName(jiraFolderName + ":admins").assignDisplayExtension
(displayExtension + " jira admins").assignDescription("Users of jira project for working group. " +
gsh_input_workingGroupDescription).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Group created: " + jiraAdminsGroup.getName());

    // Create users group
    Group jiraUsersGroup = new GroupSave().assignName(jiraFolderName + ":users").assignDisplayExtension
(displayExtension + " jira users").assignDescription("Subscribers list receives working group emails. " +
gsh_input_workingGroupDescription).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Group created: " + jiraUsersGroup.getName());

    // Add admin group to owners group
    new MembershipSave().assignGroup(jiraAdminsGroup).assignSubject(adminsGroup.toSubject()).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Added member: " + adminsGroup.getName() + " to group: " +
jiraAdminsGroup.getName());

    // Add users group to subscribers group
    new MembershipSave().assignGroup(jiraUsersGroup).assignSubject(usersGroup.toSubject()).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Added member: " + usersGroup.getName() + " to group: " +
jiraUsersGroup.getName());

    // Assign privs for admins on folder
    new PrivilegeGroupInheritanceSave().assignStem(jiraFolder).addPrivilegeName("admin").assignSubject
(jiraAdminsGroup.toSubject()).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Assigned group admin privileges to: " + jiraAdminsGroup.getName()
+ " inherited from folder: " + jiraFolder.getName());

    // Assign attestation at jira
    new AttestationStemSave().assignStem(jiraFolder).assignDaysUntilRecertify(gsh_input_attestationDays).save();
    gsh_built_in_gshTemplateOutput.addOutputLine("Assigned attestation to folder: " + jiraFolder.getName());
}

gsh_built_in_gshTemplateOutput.addOutputLine("Finished executing template: " + gsh_input_workingGroupExtension);

```

Execute with GSH

```

import edu.internet2.middleware.grouper.app.gsh.template.*;
GrouperSession.startRootSession();

GshTemplateExec exec = new GshTemplateExec();
exec.assignConfigId("createNewWorkingGroup");
Subject subject = SubjectFinder.findByIdAndSource("chris.hubing@at.internet2.edu", "ldap", true)
exec.assignCurrentUser(subject);

exec.assignGshTemplateOwnerType(GshTemplateOwnerType.stem);
exec.assignOwnerStemName("ref:incommon-collab"); // run the script from test2 folder

GshTemplateInput input = new GshTemplateInput();
input.assignName("gsh_input_workingGroupExtension");
input.assignValueString("myGroup");
exec.addGshTemplateInput(input);
input = new GshTemplateInput();
input.assignName("gsh_input_workingGroupDisplayExtension");
input.assignValueString("My group");
exec.addGshTemplateInput(input);
input = new GshTemplateInput();
input.assignName("gsh_input_workingGroupDescription");
input.assignValueString("My working group will do a lot of group work");
exec.addGshTemplateInput(input);
input = new GshTemplateInput();
input.assignName("gsh_input_isSympa");
input.assignValueString("true");
exec.addGshTemplateInput(input);
input = new GshTemplateInput();
input.assignName("gsh_input_sympaDomain");
input.assignValueString("internet2");
exec.addGshTemplateInput(input);
input = new GshTemplateInput();
input.assignName("gsh_input_isSympaModerated");
input.assignValueString("true");
exec.addGshTemplateInput(input);

// when
GshTemplateExecOutput output = exec.execute();

// then
System.out.println("Success: " + output.isSuccess());
if (!output.isSuccess() && output.getException() != null) {
    System.out.println(output.getExceptionStack());
}
System.out.println("Valid: " + output.isValid());
System.out.println("Validation:");
for (GshValidationLine gshValidationLine : output.getGshTemplateOutput().getValidationLines()) {
    System.out.println(gshValidationLine.getInputName() + ": " + gshValidationLine.getText());
}
System.out.println("Output from script:");
for (GshOutputLine gshOutputLine : output.getGshTemplateOutput().getOutputLines()) {
    System.out.println(gshOutputLine.getMessageType() + ": " + gshOutputLine.getText());
}
System.out.println("Script output:");
System.out.println(output.getGshScriptOutput());

```

Output

```
Success: true
Valid: true
Validation:
Output from script:
success: Folder created: ref:incommon-collab:myGroup
success: Group created: ref:incommon-collab:myGroup:users
success: Group created: ref:incommon-collab:myGroup:admins
success: Assigned group admin privileges to: ref:incommon-collab:myGroup:admins inherited from folder: ref:
incommon-collab:myGroup
success: Folder created: app:sympa:internet2:myGroup
success: Group created: app:sympa:internet2:myGroup:owners
success: Group created: app:sympa:internet2:myGroup:subscribers
success: Added member: ref:incommon-collab:myGroup:admins to group: app:sympa:internet2:myGroup:owners
success: Added member: ref:incommon-collab:myGroup:users to group: app:sympa:internet2:myGroup:subscribers
success: Assigned group admin privileges to: ref:incommon-collab:myGroup:admins inherited from folder: app:
sympa:internet2:myGroup
success: Finished executing template: myGroup
```

Output from Web Services Call in Internet2 Dev Environment

```
curl -H "Content-Type: text/x-json; charset=UTF-8" -d "@./createworkinggroup.json" -X POST -u comanage_provision:XXXXXXX https://grouper.dev.at.
internet2.edu/grouper-ws/servicesRest/v2\_5\_000/gshTemplateExec
```


createworkinggroup.json

```
{
  "WsRestGshTemplateExecRequest": {
    "gshTemplateActAsSubjectLookup": {
      "subjectSourceId": "ldap",
      "subjectId": "eisbruch@at.internet2.edu"
    },
    "ownerStemLookup": {
      "stemName": "ref:incommon-collab"
    },
    "ownerType": "stem",
    "configId": "createNewWorkingGroup",
    "inputs": [
      {
        "name": "gsh_input_workingGroupExtension",
        "value": "test"
      },
      {
        "name": "gsh_input_workingGroupDisplayExtension",
        "value": "Test"
      },
      {
        "name": "gsh_input_workingGroupDescription",
        "value": "This is a test."
      },
      {
        "name": "gsh_input_isSympa",
        "value": "true"
      },
      {
        "name": "gsh_input_sympaDomain",
        "value": "incommon"
      },
      {
        "name": "gsh_input_isSympaModerated",
        "value": "false"
      },
      {
        "name": "gsh_input_isOptin",
        "value": "true"
      },
      {
        "name": "gsh_input_attestationDays",
        "value": "78"
      },
      {
        "name": "gsh_input_isConfluence",
        "value": "true"
      },
      {
        "name": "gsh_input_isJira",
        "value": "true"
      }
    ]
  }
}
```

Return message

```
{
  "WsGshTemplateExecResult": {
    "resultMetadata": {
```

```

"success": "T",
"resultCode": "SUCCESS",
"resultMessage": "Success for: clientVersion: 2.5.0, configId: createNewWorkingGroup, ownerType: stem ,
inputs: Array size: 6: [0]: edu.internet2.middleware.grouper.ws.coresoap.WsGshTemplateInput@4ce20936\n[1]: edu.
internet2.middleware.grouper.ws.coresoap.WsGshTemplateInput@421fb26b\n[2]: edu.internet2.middlew...\n,
actAsSubject: null, paramNames: \n, params: null"
},
"gshScriptOutput": "groovy:001> import edu.internet2.middleware.grouper.app.gsh.template.*;\ngroovy:002>
import edu.internet2.middleware.grouper.util.*;\ngroovy:003> GshTemplateOutput gsh_builtin_gshTemplateOutput =
GshTemplateOutput.retrieveGshTemplateOutput(); \n==> edu.internet2.middleware.grouper.app.gsh.template.
GshTemplateOutput@abf4469\ngroovy:004> GshTemplateRuntime gsh_builtin_gshTemplateRuntime = GshTemplateRuntime.
retrieveGshTemplateRuntime(); \n==> edu.internet2.middleware.grouper.app.gsh.template.
GshTemplateRuntime@4aa02244\ngroovy:005> GrouperSession gsh_builtin_grouperSession =
gsh_builtin_gshTemplateRuntime.getGrouperSession();\n==>
fa3a616b1fe5426db573f6accba9ab,'GrouperSystem','application'\ngroovy:006> Subject gsh_builtin_subject =
gsh_builtin_gshTemplateRuntime.getCurrentSubject();\n==> Subject id: comanage_provision, sourceId: ldap, name:
comanage_provision\ngroovy:007> String gsh_builtin_subjectId = \"comanage_provision\";\n==>
comanage_provision\ngroovy:008> String gsh_builtin_ownerStemName = \"ref:incommon-collab\";\n==> ref:incommon-
collab\ngroovy:009> String gsh_input_workingGroupExtension = \"test\";\n==> test\ngroovy:010> String
gsh_input_workingGroupDisplayExtension = \"Test\";\n==> Test\ngroovy:011> String
gsh_input_workingGroupDescription = \"This is a test.\";\n==> This is a test.\ngroovy:012> Boolean
gsh_input_isSympa = true;\n==> true\ngroovy:013> String gsh_input_sympaDomain = \"incommon\";\n==>
incommon\ngroovy:014> Boolean gsh_input_isSympaModerated = false;\n==> false\ngroovy:015> import edu.internet2.
middleware.grouper.app.attestation.*;\ngroovy:016> import edu.internet2.middleware.grouper.attr.assign.*;
\ngroovy:017> \ngroovy:018> String workingGroupFolderName = \"ref:incommon-collab\"; +
gsh_input_workingGroupExtension;\n==> ref:incommon-collab:test\ngroovy:019> \ngroovy:020> // default value
for display extension is just the extension\ngroovy:021> String displayExtension = GrouperUtil.defaultIfBlank
(gsh_input_workingGroupDisplayExtension, gsh_input_workingGroupExtension);\n==> Test\ngroovy:022> \ngroovy:
023> // we dont want the word \"null\" printed out if this is null, so convert to empty string if this is
null\ngroovy:024> gsh_input_workingGroupDescription = GrouperUtil.defaultString
(gsh_input_workingGroupDescription);\n==> This is a test.\ngroovy:025> \ngroovy:026> // validate
first\ngroovy:027> // Check if working group folder exists\ngroovy:028> Stem workingGroupFolder = StemFinder.
findByName(gsh_builtin_grouperSession, workingGroupFolderName, false);\ngroovy:029> if (workingGroupFolder !=
null) {\ngroovy:030> gsh_builtin_gshTemplateOutput.addValidationLine(\"gsh_input_workingGroupExtension\",
\ngroovy:031> \"Error: working group extension '\" + gsh_input_workingGroupExtension + '\" already exists!
\");\ngroovy:032> \ngroovy:033> \ngroovy:034> // Check gsh_input_sympaDomain if gsh_input_isSympa\ngroovy:
035> if (gsh_input_isSympa && GrouperUtil.isBlank(gsh_input_sympaDomain)) {\ngroovy:036>
gsh_builtin_gshTemplateOutput.addValidationLine(\"gsh_input_sympaDomain\",\ngroovy:037> \"Error: Sympa
domain is required if provisioning to Sympa\";\ngroovy:038> }\ngroovy:039> \ngroovy:040> String
sympaFolderName = null;\ngroovy:041> \ngroovy:042> if (gsh_input_isSympa) {\ngroovy:043> sympFolderName =
\"app:sympa\"; + gsh_input_sympaDomain + \":\"; + gsh_input_workingGroupExtension;\ngroovy:044> Stem
sympFolder = StemFinder.findByName(gsh_builtin_grouperSession, sympFolderName, false);\ngroovy:045> if
(workingGroupFolder != null) {\ngroovy:046> gsh_builtin_gshTemplateOutput.addValidationLine(\"
gsh_input_workingGroupExtension\",\ngroovy:047> \"Error: symp folder '\" + sympFolderName + '\" already
exists!\");\ngroovy:048> }\ngroovy:049> }\ngroovy:050> \ngroovy:051> // Do not proceed if there is an
error\ngroovy:052> if (GrouperUtil.length(gsh_builtin_gshTemplateOutput.getOutputLines()) > 0) {\ngroovy:053>
gsh_builtin_gshTemplateOutput.assignIsError(true);\ngroovy:054> } else {\ngroovy:055> //
Create working group folder\ngroovy:057> Stem workingGroupFolder = new StemSave().assignName
(workingGroupFolderName).assignDisplayExtension(displayExtension)\ngroovy:058> .assignDescription(\"Folder
holds working group roles. \" + gsh_input_workingGroupDescription).save();\ngroovy:059>
gsh_builtin_gshTemplateOutput.addOutputLine(\"Folder created: \" + workingGroupFolderName);\ngroovy:060>
\ngroovy:061> // Create users group\ngroovy:062> Group usersGroup = new GroupSave().assignName
(workingGroupFolderName + \":users\")\ngroovy:063> .assignDisplayExtension(displayExtension + \" users\")
\ngroovy:064> .assignDescription(\"Users role means members of the working group with access to
collaboration tools. \" + gsh_input_workingGroupDescription).save();\ngroovy:066>
gsh_builtin_gshTemplateOutput.addOutputLine(\"Group created: \" + usersGroup.getName());\ngroovy:067>
\ngroovy:068> // Create admins group\ngroovy:069> Group adminsGroup = new GroupSave().assignName
(workingGroupFolderName + \":admins\")\ngroovy:070> .assignDisplayExtension(displayExtension + \" admins\")
\ngroovy:071> .assignDescription(\"Admins role means can manage / attest the working group. \" +
gsh_input_workingGroupDescription).save();\ngroovy:072> gsh_builtin_gshTemplateOutput.addOutputLine(\"Group
created: \" + adminsGroup.getName());\ngroovy:073> \ngroovy:074> // If the initial admin is passed
in\ngroovy:075> if (gsh_builtin_subject != null && GrouperUtil.equals(\"ldap\", gsh_builtin_subject.
getSourceId())) {\ngroovy:076> \ngroovy:077> // add the initial admin to be in the admins group\ngroovy:
078> new MembershipSave().assignGroup(adminsGroup).assignSubject(gsh_builtin_subject).save();\ngroovy:
079> gsh_builtin_gshTemplateOutput.addOutputLine(\"Added admin: \" + gsh_builtin_subject.getId() + \" to
group: \" + adminsGroup.getName());\ngroovy:080> } else {\ngroovy:081> // if not resolvable thats just a
warning\ngroovy:082> gsh_builtin_gshTemplateOutput.addOutputLine(\"info\",\ngroovy:083> \"Warning:
admin subject not resolvable or in wrong source '\" + gsh_builtin_subjectId + '\"\";\ngroovy:084> \ngroovy:
085> }\ngroovy:086> \ngroovy:087> // Assign privs for admins on folder\ngroovy:088> new
PrivilegeGroupInheritanceSave().assignStem(workingGroupFolder)\ngroovy:089> .addPrivilegeName(\"admin\").
assignSubject(adminsGroup.toSubject()).save();\ngroovy:090> gsh_builtin_gshTemplateOutput.addOutputLine(\"

```

```

Assigned group admin privileges to: \" + adminsGroup.getName() + \" inherited from folder: \" +
workingGroupFolder.getName());\ngroovy:091> \ngroovy:092> // Assign attestation on working group
folder\ngroovy:093> new AttestationStemSave().assignStemName(workingGroupFolderName).assignDaysUntilRecertify
(90).save();\ngroovy:094> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Assigned attestation to folder: \" +
workingGroupFolderName);\ngroovy:095> \ngroovy:096> // sympa folder\ngroovy:097> if (gsh_input_isSympa)
{\ngroovy:098> Stem sympaFolder = new StemSave().assignName(sympaFolderName).assignDisplayExtension
(displayExtension + \" sympa\")\ngroovy:099> .assignDescription(\"Folder email lists for working group.
\" + gsh_input_workingGroupDescription).save();\ngroovy:100> gsh_builtIn_gshTemplateOutput.addOutputLine(\"
Folder created: \" + sympaFolder.getName());\ngroovy:101> \ngroovy:102> // Create admins group\ngroovy:
103> Group ownersGroup = new GroupSave().assignName(sympaFolderName + \":owners\")\ngroovy:104> .
assignDisplayExtension(displayExtension + \" sympa owners\")\ngroovy:105> .assignDescription(\"Owners
list manages the email list for the working group. \" + gsh_input_workingGroupDescription).save();\ngroovy:
106> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Group created: \" + ownersGroup.getName());\ngroovy:
107> \ngroovy:108> // Create subscribers group\ngroovy:109> Group subscribersGroup = new GroupSave().
assignName(sympaFolderName + \":subscribers\")\ngroovy:110> .assignDisplayExtension(displayExtension + \"
sympa subscribers\")\ngroovy:111> .assignDescription(\"Subscribers list receives working group emails.
\" + gsh_input_workingGroupDescription).save();\ngroovy:112> gsh_builtIn_gshTemplateOutput.addOutputLine(\"
Group created: \" + subscribersGroup.getName());\ngroovy:113> \ngroovy:114> // Add admin group to owners
group\ngroovy:115> new MembershipSave().assignGroup(ownersGroup).assignSubject(adminsGroup.toSubject()).
save();\ngroovy:116> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Added member: \" + adminsGroup.getName()
+ \" to group: \" + ownersGroup.getName());\ngroovy:117> \ngroovy:118> // Add users group to subscribers
group\ngroovy:119> new MembershipSave().assignGroup(subscribersGroup).assignSubject(usersGroup.toSubject()).
save();\ngroovy:120> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Added member: \" + usersGroup.getName()
+ \" to group: \" + subscribersGroup.getName());\ngroovy:121> \ngroovy:122> // Assign privs for admins on
folder\ngroovy:123> new PrivilegeGroupInheritanceSave().assignStem(sympaFolder)\ngroovy:124> .
addPrivilegeName(\"admin\").assignSubject(adminsGroup.toSubject()).save();\ngroovy:125>
gsh_builtIn_gshTemplateOutput.addOutputLine(\"Assigned group admin privileges to: \" + adminsGroup.getName() +
\" inherited from folder: \" + sympaFolder.getName());\ngroovy:126> \ngroovy:127> // Add moderated marker
attribute to sympa folder if gsh_input_isSympa and gsh_input_isSympaModerated\ngroovy:128> if
(gsh_input_isSympaModerated) {\ngroovy:129> \ngroovy:130> // Assign a sympa attribute to the
sympa folder for midpoint\ngroovy:131> new AttributeAssignToStemSave().assignStem(sympaFolder).
assignNameOrAttributeDefName(\"app:sympa:attribute:moderated\").save();\ngroovy:132>
gsh_builtIn_gshTemplateOutput.addOutputLine(\"Assigned attribute: app:sympa:attribute:moderated to: \" +
sympaFolder.getName());\ngroovy:133> }\ngroovy:134> \ngroovy:135> // Assign attestation at
sympa\ngroovy:136> new AttestationStemSave().assignStem(sympaFolder).assignDaysUntilRecertify(90).save();
\ngroovy:137> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Assigned attestation to folder: \" +
sympaFolder.getName());\ngroovy:138> \ngroovy:139> }\ngroovy:140> }n==== edu.internet2.middleware.grouper.
app.gsh.template.GshTemplateOutput@abf4469\ngroovy:141> gsh_builtIn_gshTemplateOutput.addOutputLine(\"Finished
executing template: \" + gsh_input_workingGroupExtension);\n==== edu.internet2.middleware.grouper.app.gsh.
template.GshTemplateOutput@abf4469\",
"gshOutputLines": [
{
"messageType": "success",
"text": "Folder created: ref:incommon-collab:test"
},
{
"messageType": "success",
"text": "Group created: ref:incommon-collab:test:users"
},
{
"messageType": "success",
"text": "Group created: ref:incommon-collab:test:admins"
},
{
"messageType": "success",
"text": "Added admin: comanage_provision to group: ref:incommon-collab:test:admins"
},
{
"messageType": "success",
"text": "Assigned group admin privileges to: ref:incommon-collab:test:admins inherited from folder: ref:
incommon-collab:test"
},
{
"messageType": "success",
"text": "Assigned attestation to folder: ref:incommon-collab:test"
},
{
"messageType": "success",
"text": "Folder created: app:sympa:incommon:test"
}
]
}

```

```

        "messageType": "success",
        "text": "Group created: app:sympa:incommon:test:owners"
    },
    {
        "messageType": "success",
        "text": "Group created: app:sympa:incommon:test:subscribers"
    },
    {
        "messageType": "success",
        "text": "Added member: ref:incommon-collab:test:admins to group: app:sympa:incommon:test:owners"
    },
    {
        "messageType": "success",
        "text": "Added member: ref:incommon-collab:test:users to group: app:sympa:incommon:test:subscribers"
    },
    {
        "messageType": "success",
        "text": "Assigned group admin privileges to: ref:incommon-collab:test:admins inherited from folder: app:sympa:incommon:test"
    },
    {
        "messageType": "success",
        "text": "Assigned attestation to folder: app:sympa:incommon:test"
    },
    {
        "messageType": "success",
        "text": "Finished executing template: test"
    }
],
"responseMetadata": {
    "serverVersion": "2.5.44",
    "millis": "28553"
},
"gshValidationLines": [],
"transaction": false
}
}

```