HOWTO Integrate MediaWiki with Grouper on Ubuntu Linux 14.04

19 November 2014

Author: Marco Malavolti, Andrea Biancini

Index

1 Introduction

This is a tutorial for users that want integrate the authentication for a MediaWiki site with Grouper on an Service Provider based on Ubuntu Linux 14.04.

2 Requirements

- A machine with Grouper installed and configured.
- A Shibboleth SP (sp.example.it) with Apache2 Web Server, PHP 5.3.2 or later and MySQL 5.0.2 or later.
- MediaWiki plugin

3 Installation

Install and Configure MediaWiki on your Service Provider

1. Install the software requirements for MediaWiki on your SP:
   - `apt-get install apache2 mysql-server php5-mysql phpmyadmin php5-intl php-apc`

2. Install MediaWiki on your SP:
   - `sudo su -`
   - `cd /usr/local/src`
   - `wget http://releases.wikimedia.org/mediawiki/1.23/mediawiki-1.23.6.tar.gz`
   - `tar xzf mediawiki-1.23.6`
   - `mv mediawiki-1.23.6 /var/www`
   - `vim /etc/apache2/sites-available/mediawiki.conf`:

```
<IfModule mod_alias.c>
  Alias /wiki /var/www/mediawiki-1.23.6/

  <Directory /var/www/mediawiki-1.23.6/>
    Options Indexes MultiViews FollowSymLinks
    Order deny,allow
    Allow from all
  </Directory>

</IfModule>
```

   - `a2ensite mediawiki.conf ; service apache2 reload`
   - `mysql -u root -p (type your mySQL root password to access)`
     - `create database wikidb;`
     - `grant index, create, select, insert, update, delete, alter, lock tables on wikidb.* to 'wikiuser'@'localhost' identified by '##wikiuserPassword_CHANGE_ME##';`
     - `flush privileges;`
     - `quit`

3. Configure MediaWiki on your SP:
   - Open the page "**https://sp.example.it/wiki/mw-config/index.php**" and follow the instructions until the end.
   - Download the "**LocalSettings.php**" file and put it into the MediaWiki directory (`/var/www/mediawiki-1.23.6/`)

4 Install and Configure Shibboleth plugin for MediaWiki

The following instruction is based on the page:
http://www.mediawiki.org/wiki/Extension:Shibboleth_Authentication

1. Configure Shibboleth SP to support the MediaWiki authentication plugin:
   - `vim /etc/shibboleth/shibboleth2.xml`

```
<Host name="sp.example.it">
  <Path name="wiki" authType="shibboleth" requireSession="false"/>
</Host>
```

```
<SessionInitiator type="Chaining" Location="/DS" id="DS" relayState="cookie"
isDefault="true">
  <SessionInitiator type="SAML2" acsIndex="1" template="bindingTemplate.html"/>
  <SessionInitiator type="Shib1" acsIndex="5"/>
  <SessionInitiator type="WAYF" acsIndex="5" URL="https://dsc.example.it/DS"/>
</SessionInitiator>
```

   - `shibd -t /etc/shibboleth/shibboleth2.xml`
   - `service shibd restart`

2. Edit the apache2 wiki configuration to support the Shibboleth authentication:
   - `vim /etc/apache2/sites-enabled/mediawiki.conf`:

```
<IfModule mod_alias.c>
  Alias /wiki /var/www/mediawiki-1.23.6/

  <Directory /var/www/mediawiki-1.23.6/>
    Options Indexes MultiViews FollowSymLinks
    Order deny,allow
    Allow from all
  </Directory>

  <Location /wiki>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-attr entitlement urn:mace:garr.it:wiki
  </Location>

</IfModule>
```

(This rules are compatible with apache 2.4 or later.
If you use a previous apache version, please, use this Location:

```
  <Location /wiki>
    AuthType shibboleth
    ShibRequireSession On
    require entitlement urn:mace:garr.it:wiki
  </Location>
```

)

3. Copy the extension code provide by THIS PAGE into **/var/www/mediawiki-1.23.6/extensions/ShibAuthPlugin.php**

4. Modify the "**/var/www/mediawiki-1.23.6/LocalSettings.php**" by adding these lines:

```
# The following permissions were set based on your choice in the installer
$wgGroupPermissions['*']['edit'] = false;


# Add more configuration options below.
// Shibboleth Authentication Stuff
// Load ShibAuthPlugin
require_once('extensions/ShibAuthPlugin.php');


// Last portion of the shibboleth WAYF url for lazy sessions.
// This value is found in your shibboleth.xml file on the setup for your SP
// WAYF url will look something like: /Shibboleth.sso/WAYF/$shib_WAYF
$shib_WAYF = "DS";


// Are you using an old style WAYF (Shib 1.3)
// or new style Discover Service (Shib 2.x)?
// Values are WAYF or DS, defaults to WAYF
$shib_WAYFStyle = "DS";


// Is the assertion consumer service located
// at an https address (highly recommended)
// Default for compatibility with previous version: false
$shib_Https = true;


// Prompt for user to login
$shib_LoginHint = "Login via Single Sign-on";


// Where is the assertion consumer service located on the website?
// Default: "/Shibboleth.sso"
$shib_AssertionConsumerServiceURL = "/Shibboleth.sso";


// Map Real Name to what Shibboleth variable(s)?
# $shib_RN = ucfirst(strtolower($_SERVER['HTTP_FIRST_NAME'])) . ' '
#     . ucfirst(strtolower($_SERVER['HTTP_LAST_NAME']));
$shib_RN = $_SERVER['cn'];


// Map e-mail to what Shibboleth variable?
$shib_email = $_SERVER['mail'];


// Field containing groups for the user
if (array_key_exists("isMemberOf", $_SERVER)) {
        $shib_groups = $_SERVER['isMemberOf'];
}

//This value must match with the FolderID of Wiki on the Grouper instance
$shib_group_prefix = "wiki";
```

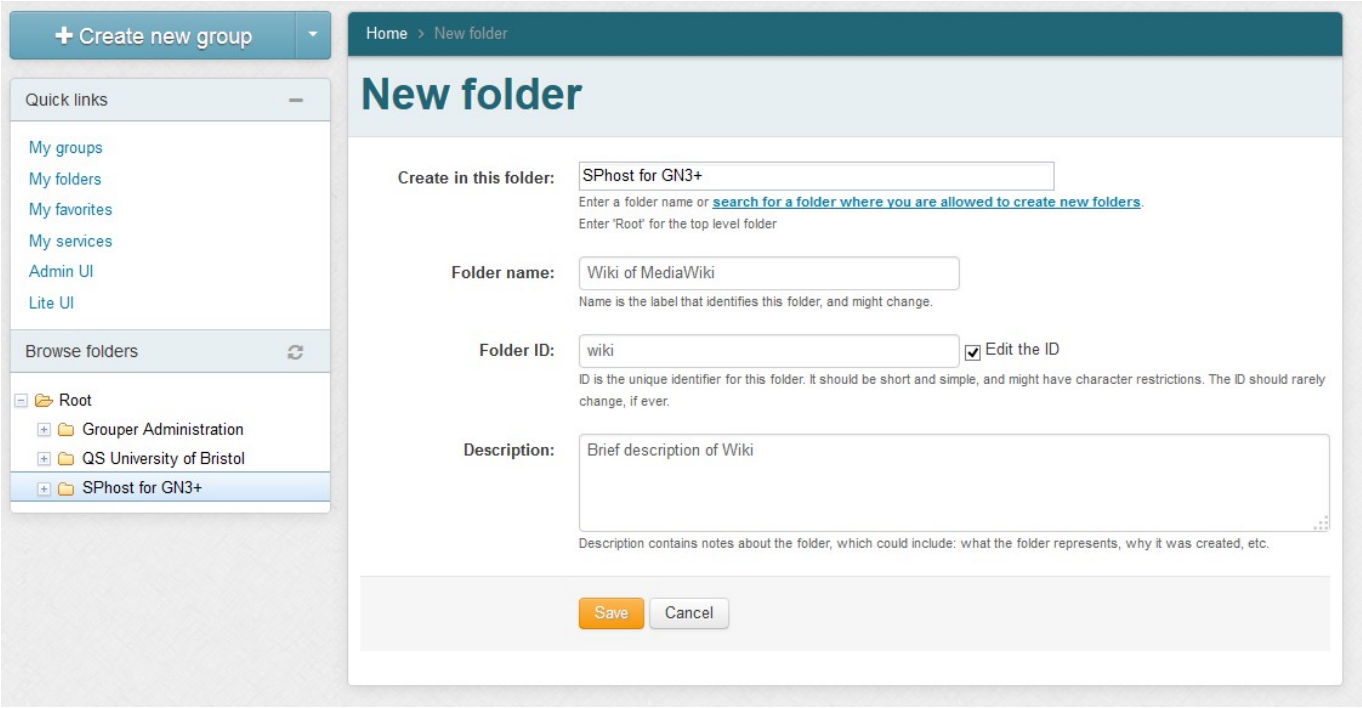
```
// The ShibUpdateUser hook is executed on login.
// It has two arguments:
// - $existing: True if this is an existing user, false if it is a new user being
added
// - &$user: A reference to the user object.
//           $user->updateUser() is called after the function finishes.
// In the event handler you can change the user object, for instance set the email
address or the real name
// The example function shown here should match behavior from previous versions of
the extension:

$wgHooks['ShibUpdateUser'][] = 'ShibUpdateTheUser';

function ShibUpdateTheUser($existing, $user) {
     global $shib_email;
     global $shib_RN;
     if (! $existing) {
          if($shib_email != null)
               $user->setEmail($shib_email);
          if($shib_RN != null)
               $user->setRealName($shib_RN);
     }
}

// This is required to map to something
// You should beware of possible namespace collisions, it is best to chose
// something that will not violate MW's usual restrictions on characters
// Map Username to what Shibboleth variable?
$shib_UN = ucfirst(strtolower($_SERVER['eppn']));


// Shibboleth doesn't really support logging out very well.  To take care of
// this we simply get rid of the logout link when a user is logged in through
// Shib.  Alternatively, you can uncomment and set the variable below to a link
// that will either clear the user's cookies or log the user out of the Idp and
// instead of deleting the logout link, the extension will change it instead.
$shib_logout = "/Shibboleth.sso/Logout";

//Add to permit the management of the User rights
$wgUserrightsInterwikiDelimiter = '#';


// Activate Shibboleth Plugin
SetupShibAuth();
```

5. Modify the value "**protected mRemember**" to "**var mRemember**" into the "**/var/www/mediawiki-1.23.6/includes/specials/SpecialUserlogin.php**" file.

6. Modify the "**attribute-filter.xml**" of your Identity Provider and restart its Tomcat server to release the attributes "**commonName**", "**eduPersonPrincipalName**" and "**mail**" to this SP.

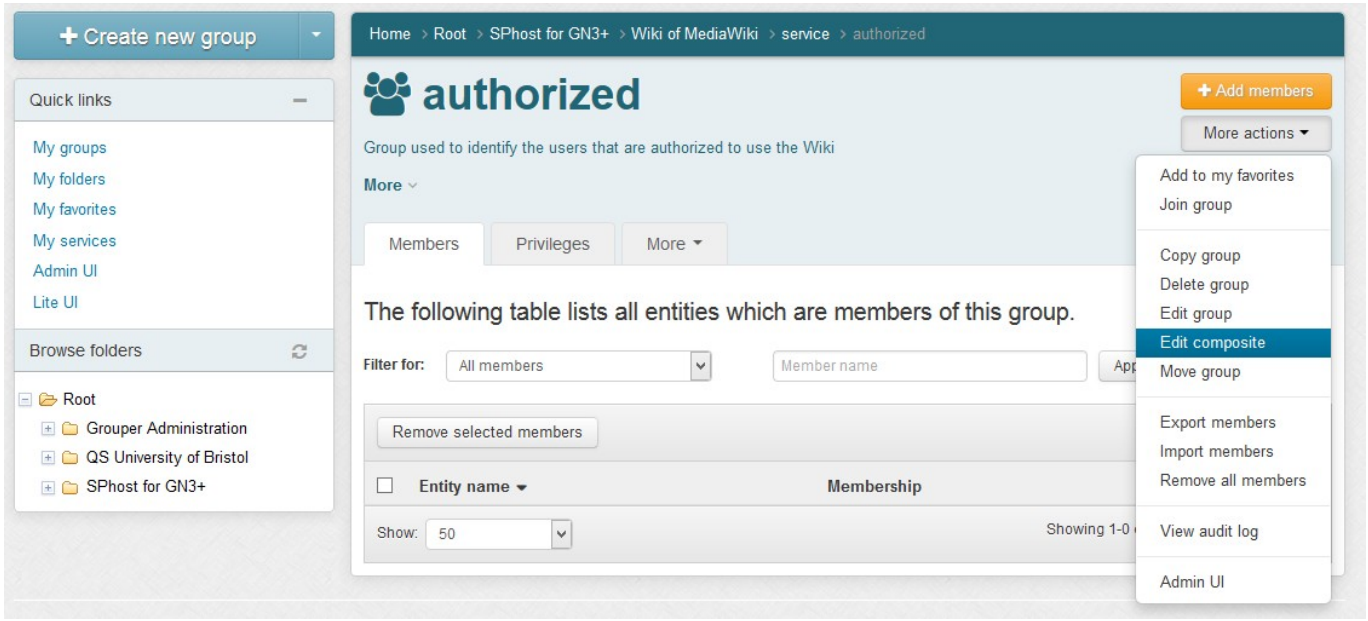
5 Add the MediaWiki groups to Grouper

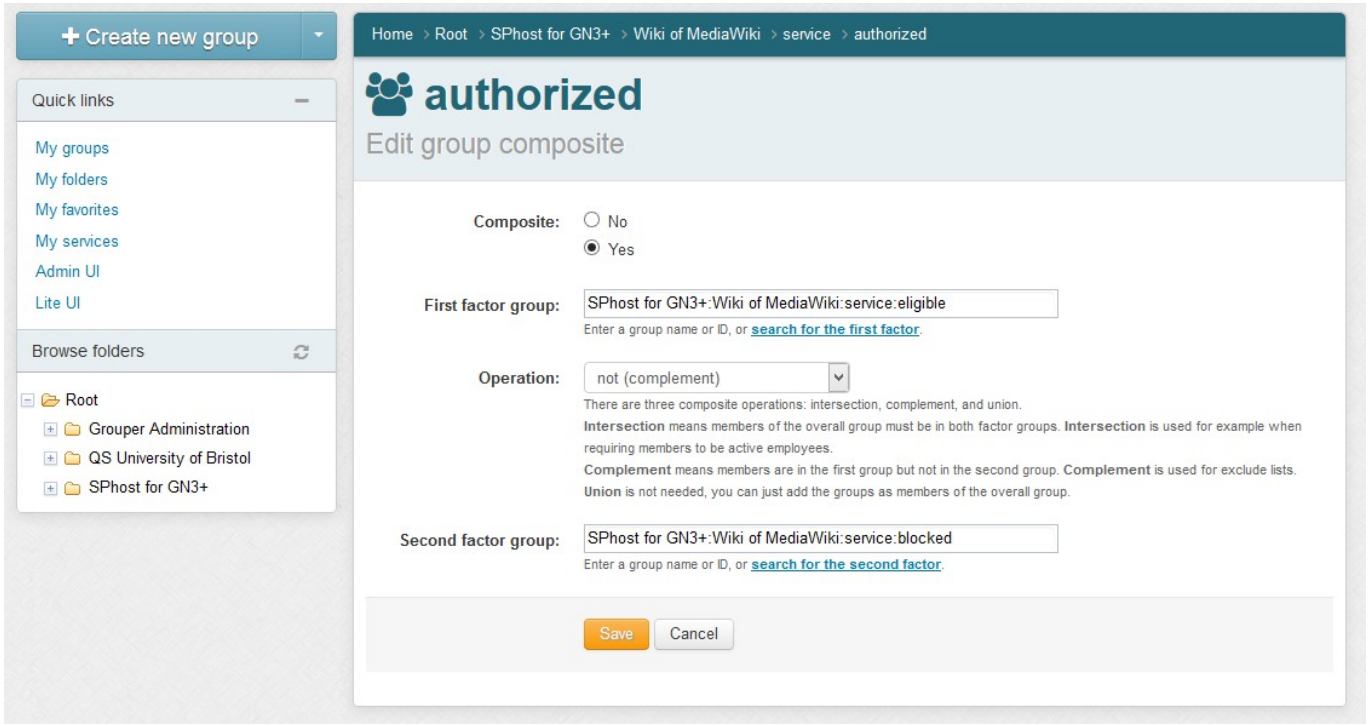
1. Open Grouper application, log-in as Admin and create the folder for the WiKi ("WiKi of MediaWiki" with Folder ID = "**wiki**") under the relying party's folder:



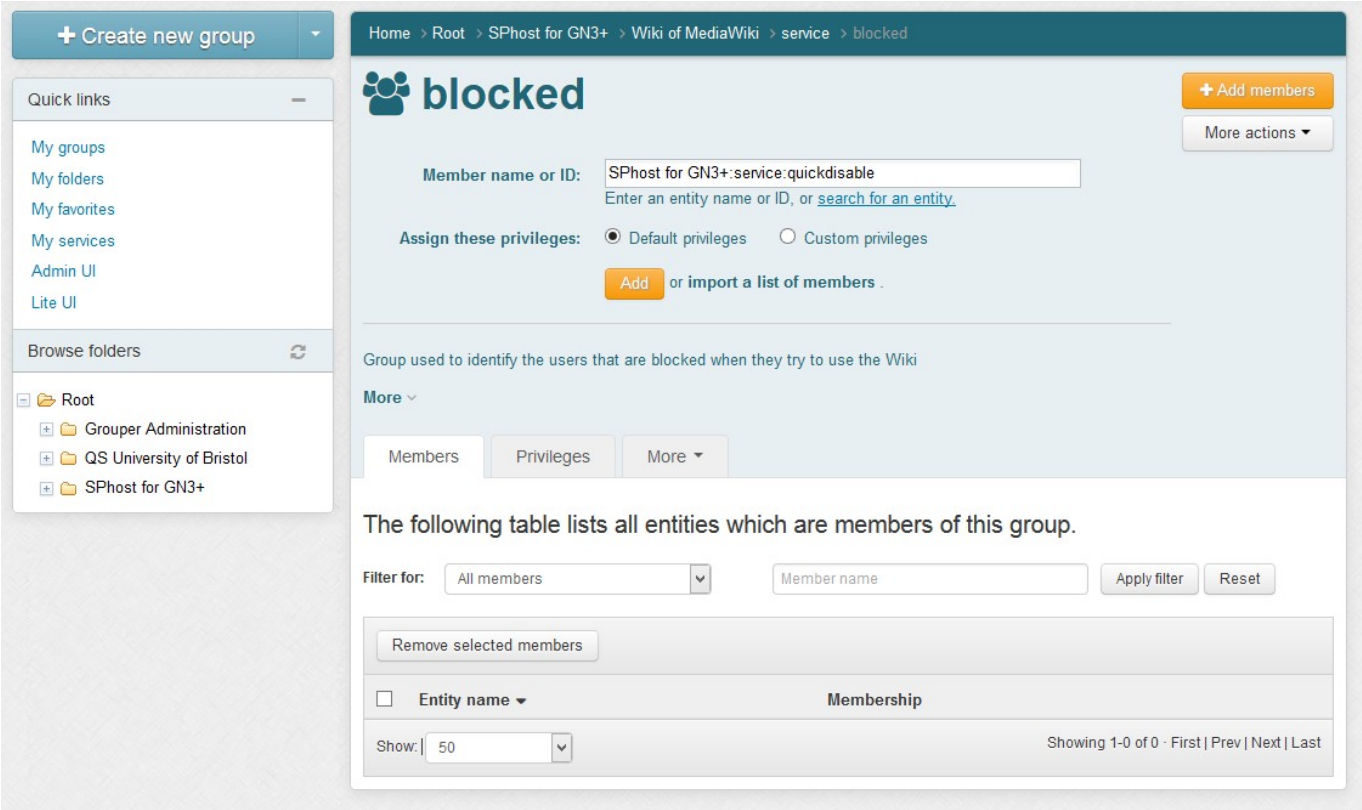
2. Create the "**service**" folder into the "**wiki**" folder and add to it these groups:
   a) authorized group:
      - Name: **auhtorized**
      - ID: **authorized**
      - Description: **Group used to identify the users that are authorized to use the Wiki**
   b) eligible group:
      - Name: **eligible**
      - ID: **eligible**
      - Description: **Group used to identify the users that are eligible to use the Wiki**
   c) blocked group:
      - Name: **blocked**
      - ID: **blocked**
      - Description: **Group used to identify the users that are blocked when they try to use the Wiki**

3. Make the "**authorized**" group a composite group that involves "**eligible**" and "**blocked**" groups:





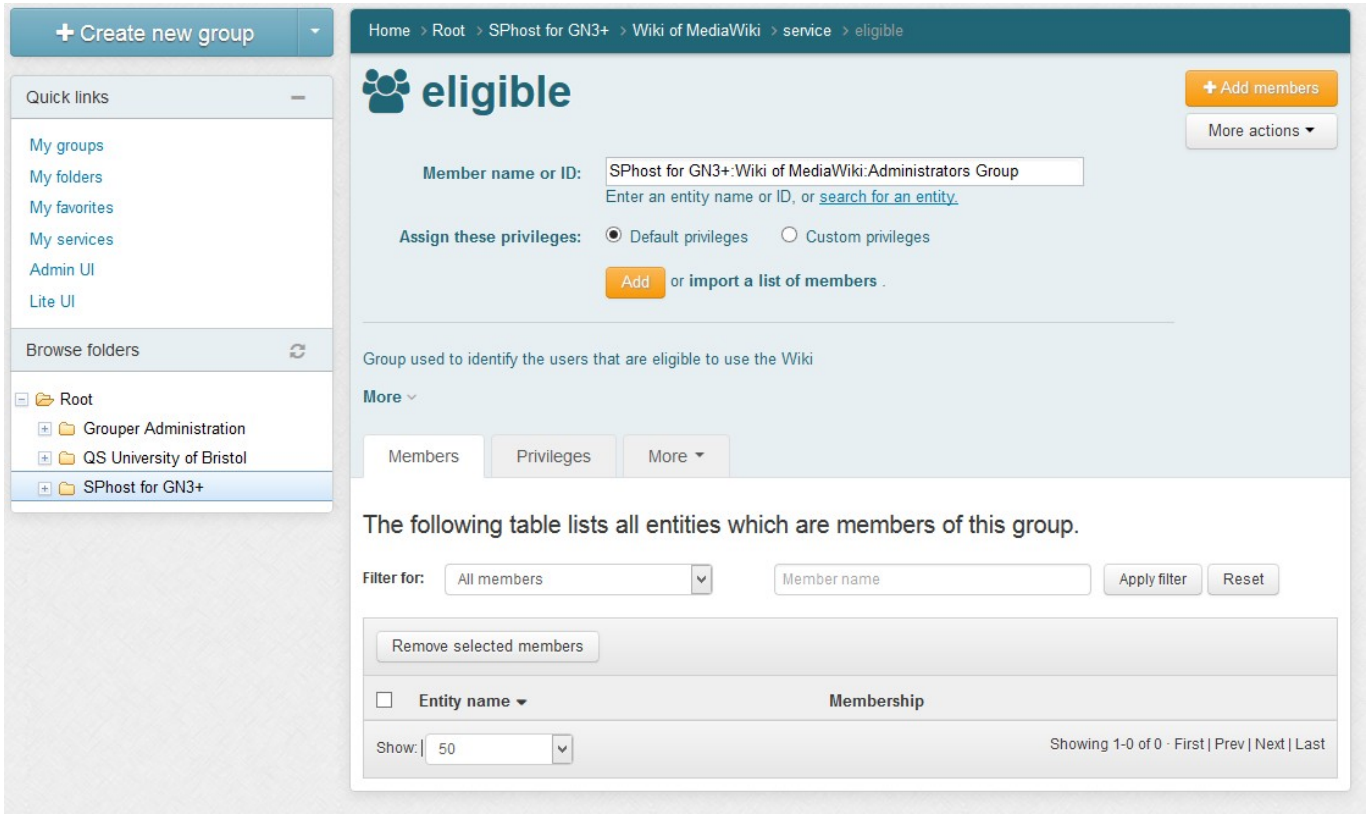
4. Add the group designed to contain the users that are blocked on all services hosted by the SP to the "**blocked**" group:



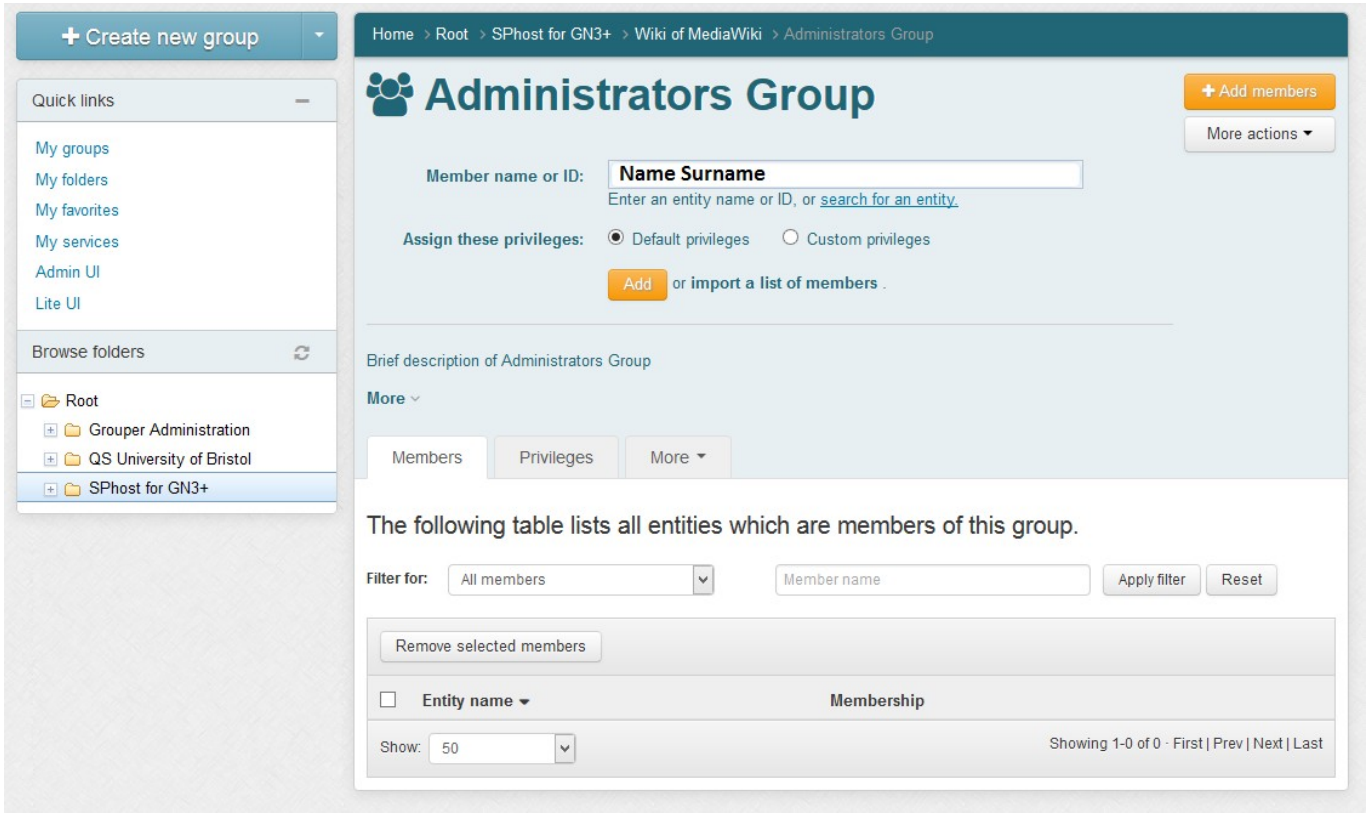
5. Into the main "**wiki**" folder create these new groups:
   a) Administrators Group:
      - Name: **Administrators Group**
      - ID: **sysop**
      - Description: **Brief description of Administrators Group**
   b) Bureaucrats Group:
      - Name: **Bureaucrats Group**
      - ID: **bureaucrat**
      - Description: **Brief description of Bureaucrats Group**
   c) Users Group:
      - Name: **Users Group**
      - ID: **user**
      - Description: **Brief description of Users Group**

(each ID value matches with the default groups of MediaWiki software and stored into its DB)

6. Add the new groups created to the "**eligible**" group of "**wiki**", for example:



7. Add the members that you want elevate to administrators of Wiki to the *Administrators Group* and to *Bureaucrats Group*. For Example:



8. Try to access into the Wiki and navigate through the special pages to understand if your user is administrator or not.
   If you are Administrator of the Wiki, you can find the page "**User Rights Management**" between the special pages.

6 Add new MediaWiki groups and link them to Grouper

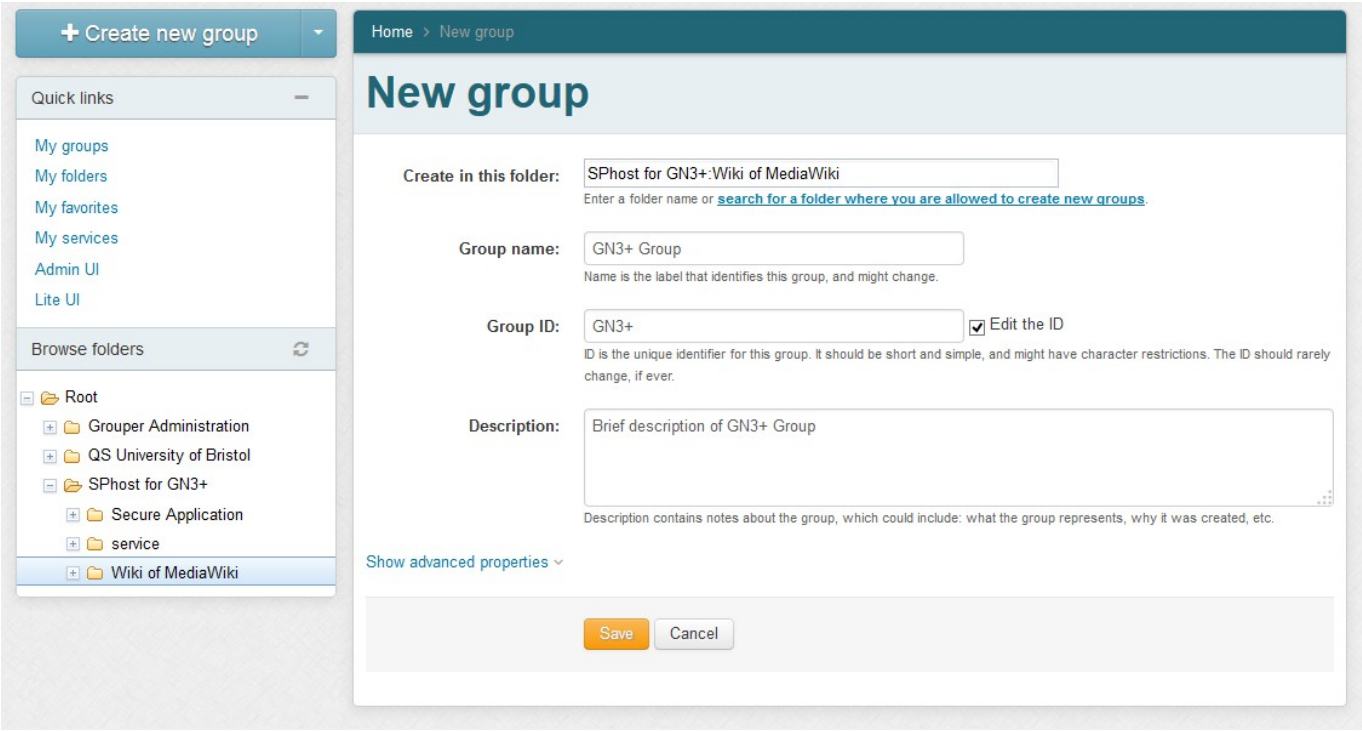
1. Modify the "**/var/www/mediawiki-1.23.6/LocalSettings.php**" by adding these lines:

```
// New GN3+ group and its permissions
$wgGroupPermissions['GN3+']['move'] = true;
$wgGroupPermissions['GN3+']['move-subpages'] = true;
$wgGroupPermissions['GN3+']['move-rootuserpages'] = false;
$wgGroupPermissions['GN3+']['movefile'] = true;
$wgGroupPermissions['GN3+']['read'] = true;
$wgGroupPermissions['GN3+']['edit'] = true;
$wgGroupPermissions['GN3+']['createpage'] = true;
$wgGroupPermissions['GN3+']['createtalk'] = true;
$wgGroupPermissions['GN3+']['writeapi'] = false;
$wgGroupPermissions['GN3+']['upload'] = true;
$wgGroupPermissions['GN3+']['reupload'] = true;
$wgGroupPermissions['GN3+']['reupload-shared'] = true;
$wgGroupPermissions['GN3+']['minoredit'] = true;
$wgGroupPermissions['GN3+']['purge'] = false;
$wgGroupPermissions['GN3+']['sendemail'] = false;

// Activate Shibboleth Plugin
SetupShibAuth();
```

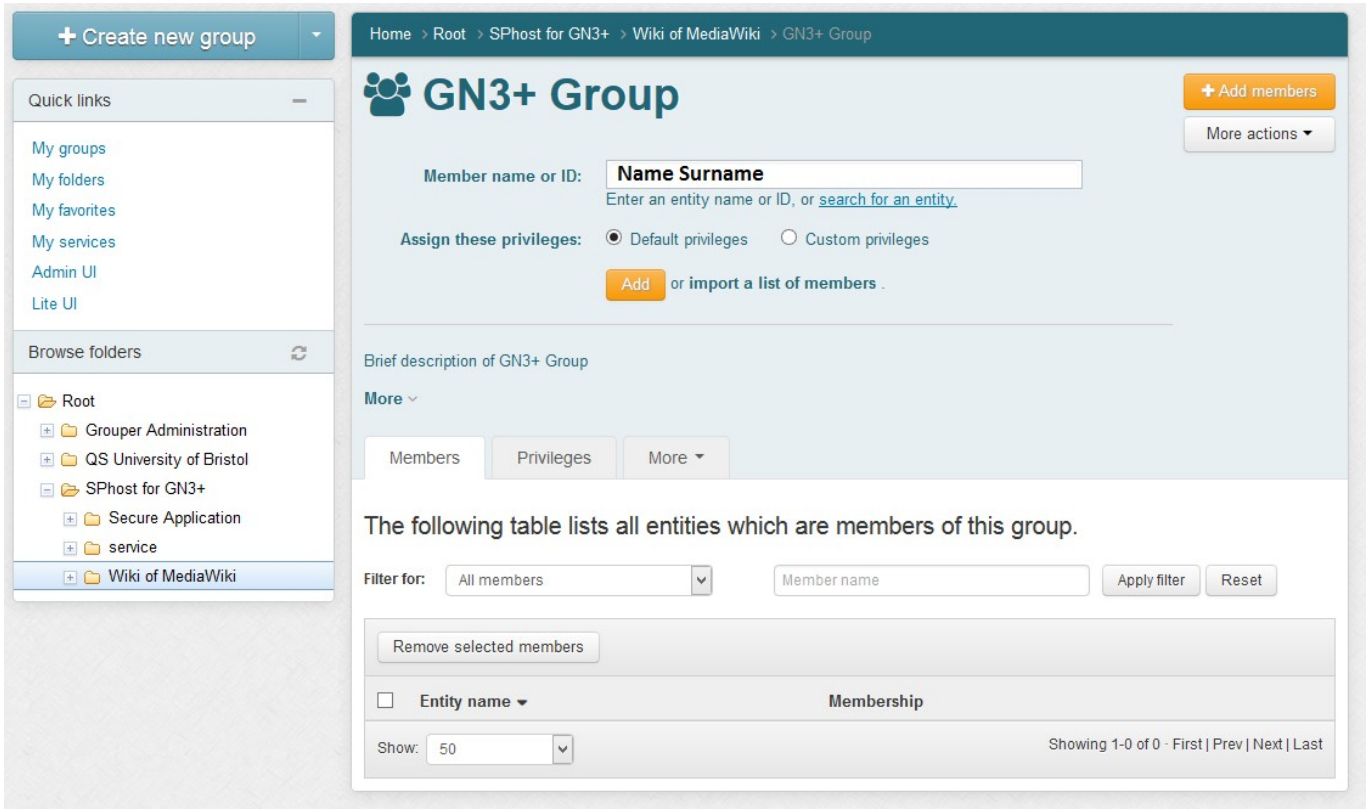
(Each line defines each permission that the "**GN3+**" group will have. See the MediaWiki User Rights [2])

2. Move on your Grouper instance and:
   a) Create this new GN3+ group under the "**wiki**" folder:
      - Name: **GN3+ Group**
      - ID: **GN3+**
      - Description: **Brief description of GN3+ Group**

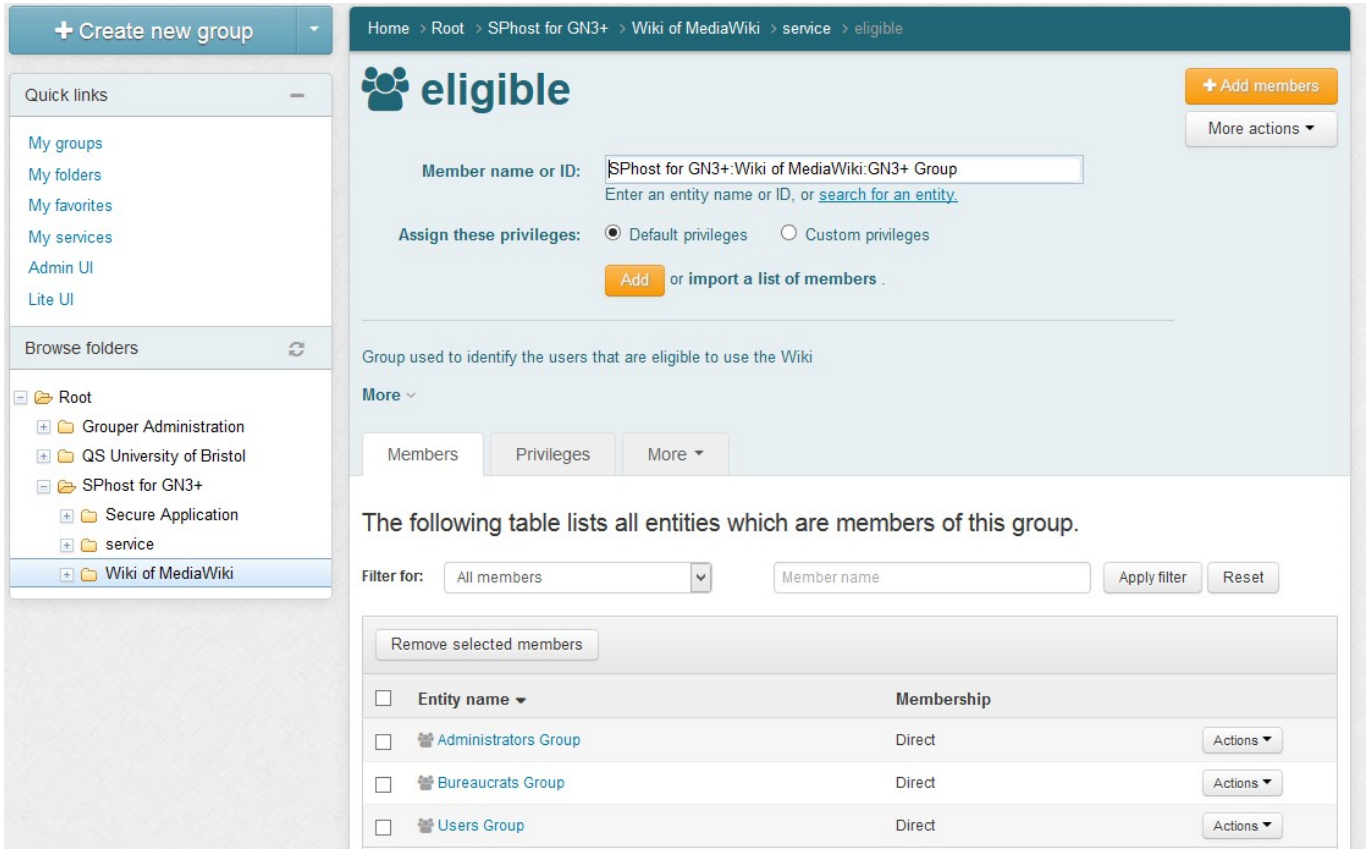
   (the value of "ID" must match with the name of the new MediaWiki group)



b) Add the members that you want to be part of this new GN3+ group.



c) Add the new "**GN3+**" group to the "**eligible**" group under "**service**" folder:



3. Access to the Wiki and navigate through the special page "**User group rights**" and find the new MediaWiki group called "GN3+".
4. Verify that the GN3+ group has all its members by open the link called "*(list of members)*".

7 How to prevent the access to an user

1. Add the user that you want block to the "**blocked**" group on the Grouper instance.
2. After that, when the user will try to access on the Wiki, he will meet the **401 Unauthorized** error page.

8 Bibliography

[1] MediaWiki: https://www.mediawiki.org/wiki/MediaWiki

[2] MediaWiki User Rights: http://www.mediawiki.org/wiki/Manual:User_rights