Mediawiki Extension Restrict Access By Category and Group

Mediawiki Extension Restrict Access By Category and Group

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-04-14
Document type:	Documentation
Status:	Active
Access:	free

□

Contents

- 1 Description
- 2 Usage
- 3 Download instructions
- 4 Installation
  - 4.1 Folder creation
  - 4.2 Create the extension php file
  - 4.3 Edit Localsettings.php
    - 4.3.1 Activate the extension
    - 4.3.2 Add the security groups

Description

This extension can restrict access to users by group and document category.

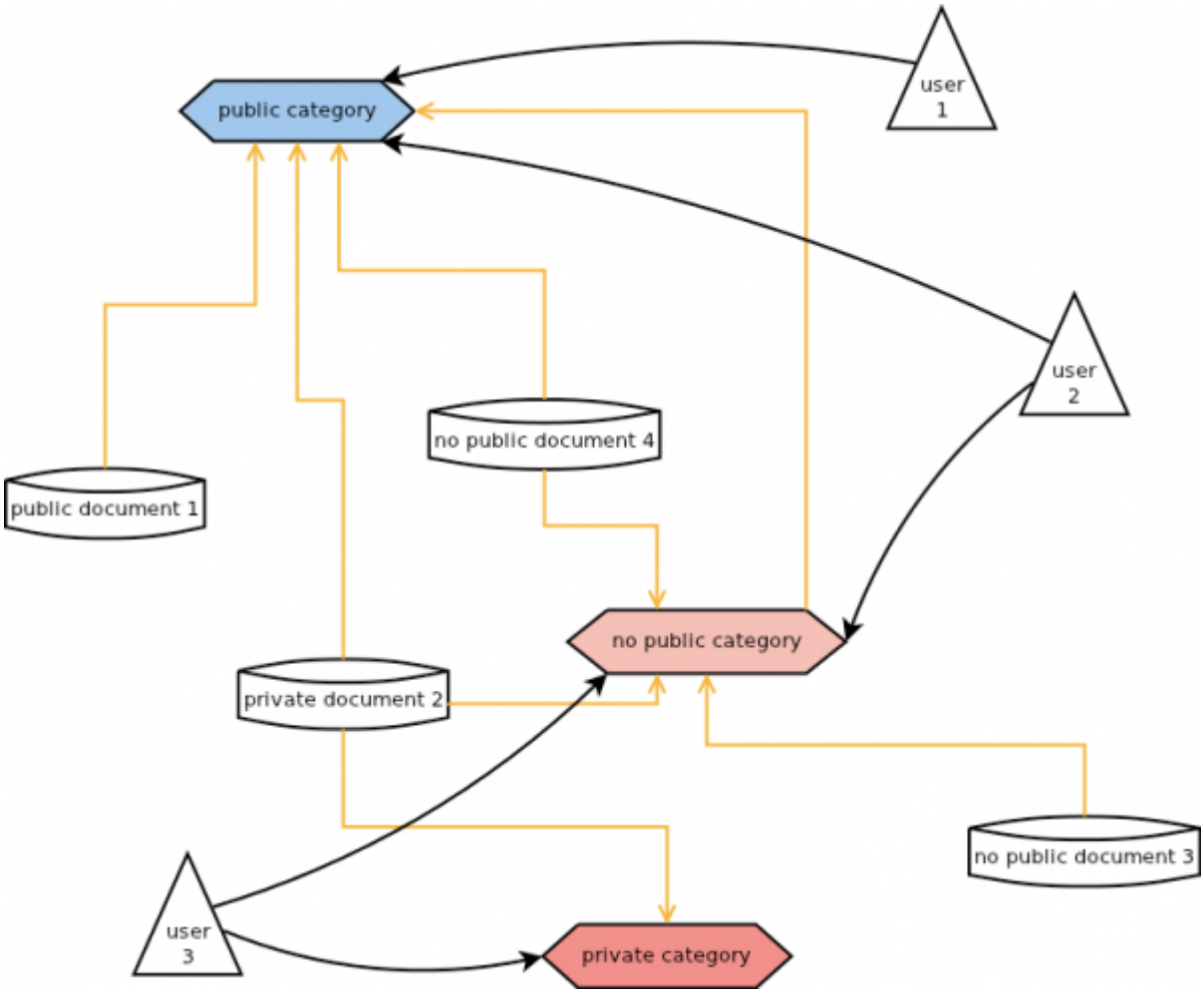
Even though Mediawiki is a free/public access collaborative document tool, sometimes it can be helpful, especially in business environments, to have a restricted view of wiki documents. Especially for a private wiki.

For example, department's financial documents should not be accessed by customer services users.

In this extension you can establish four access restrictions:

- Public: White pages: those are public pages that can be accessed by everybody. This is helpful when you have a private mediawiki and anonymous can only authenticate and see Main page. See $wgWhitelistRead.
- Public categories: those are all categories that aren't in your groups.php file. Those categories are public.

- No public categories: those are all categories that are in your groups.php file. Those categories are restricted. Documents that belong to these categories may be accessed by users who belong to at least one of these groups.
- Private categories:those are all categories that are in your groups.php file with [private] = true option. These categories are private, and only users who belong to one or more of this categories closed to which the document will have access. It is the same behavior that if the document did not belong to any of the no public categories basis only membership to the private category (respect to access).



- user1 could access to public document 1 only.
- user 2 could access to public document 1, no public document 3 and no public document 4.
- user 3 could access to public document 1, no public document 3, no public document 4 and private document 2.

Usage

All documents you want to restrict access to, you must add to a category. So you have multiple categories.

Files:

- $IP/extensions/rabcg/rabcg.php: this is the extension.
- $IP/extensions/rabcg/groups.php: this is the group catalog.

In your groups.php file, you must add the categories you want to make no public or private. This is made by group definition.
For Example:

```
<?php
// This is a no public category: Financial no public data.
$wgGroupPermissions['Financial no public data']['*'] = true;
// This is a private category: Financial private data.
$wgGroupPermissions['Financial private data']['private'] = true;
```

To apply this category to your document, you only must write:

This is only one more category of your document.

Previously (after create the groups in you groups.php file), you must make groups assignments to users by Special:UserRights page.

Download instructions

This extension is not yet in MediaWiki SVN Repository.

Therefore, you must copy & paste the following code as is explained in the installation section below.

Installation

Folder creation

Create a folder in your root/extensions folder :

```
sudo mkdir RestrictAccessByCategoryAndGroup
```

Create the extension php file

```
sudo nano
RestrictAccessByCategoryAndGroup/RestrictAccessByCategoryAndGroup.php
```

And insert the following code:

```
<?php
if ( !defined( 'MEDIAWIKI' ) ) {
        die( 'Not a valid entry point.' );
}
$wgExtensionCredits['parserhook'][] = array(
        'name' => 'Restrict access by category and group',
        'author' => 'Andrés Orencio Ramirez Perez',
        'url' =>
'https://www.mediawiki.org/wiki/Extension:Restrict_access_by_category_and_gro
```

```
up',
        'description' => 'Allows to restrict access to pages by users groups
and page categories',
        'version' => '2.0.1'
);
$wgHooks['userCan'][] = 'restrictAccessByCategoryAndGroup';
function restrictAccessByCategoryAndGroup( $title, $user, $action, $result )
{
        global $wgGroupPermissions;
        global $wgWhitelistRead;
        global $wgLang;
        global $wgHooks;
        global $wgContLang;
        global $wgWhitelistRead;
        global $wgVersion;
//The Main Page, Login and Logout pages should always be accessible
        if ( $wgVersion >= '1.17' ) {
                $wgWhitelistRead[] = wfMessage( 'mainpage' )->plain();
        } else {
                $wgWhitelistRead[] = wfMsgForContent( 'mainpage' );
        }
        $wgWhitelistRead[] = SpecialPage::getTitleFor( 'Userlogin'
)->getLocalUrl();
        $wgWhitelistRead[] = SpecialPage::getTitleFor( 'Userlogout'
)->getLocalUrl();
$validCategory = false;
        $groupExists = false;
        $pageHasCategories = false;
        $privateCategory = false;
        $privateCategoryTemp = false;
        $categoryNamespace = $wgLang->getNsText( NS_CATEGORY );
        $whitePage = true;
//System categories
        $systemCategory = array();
        foreach ( array_change_key_case( $title->getParentCategories(),
CASE_LOWER ) as $key => $value ) {
                $formatedKey = substr( $key, ( strpos( $key, ":" ) + 1 ) );
                $systemCategory[ $formatedKey ] = $value;
        }
//Is this page a white page?
        if ( isset( $wgWhitelistRead[0] ) ) {
                $whitePage = in_array( $title, $wgWhitelistRead );
        }
//If the page has no categories, it's public.
        if ( count( $title->getParentCategories() ) == 0 ) {
                $validCategory = true;
        } else {
                //For each system categories
                foreach ( $wgGroupPermissions as $key => $value ) {
                        //If current system category is defined as private,
then tmpCatP is true
```

```
                        if ( isset( $wgGroupPermissions[ $key ]['private'] )
) {
                                $privateCategoryTemp = $wgGroupPermissions[
$key ]['private'];
                        } else {
                                $privateCategoryTemp = false;
                        }
                        //If current system category exist in the document
category array ...
                        if ( array_key_exists( strtolower( str_replace( " ",
"_", $key ) ), $systemCategory ) ) {
                                if ( $privateCategoryTemp
and !$privateCategory ) {
                                        $privateCategory = true;
                                        $validCategory = false;
                                }
                                //We see that the user belongs to one of the
groups (like of category)
                                if ( in_array( $key, $user->getGroups() ) and
( !$privateCategory or ( $privateCategoryTemp and $privateCategory ) ) ) {
                                        $validCategory = true;
                                }
                                $groupExists = true;
                        }
                }
                $pageHasCategories = count( $title->getParentCategories() ) >
0;
        }
if ( !$pageHasCategories ) {
                return true;
        }
        if ( !$groupExists and !$whitePage ) {
                return true;
        }
        if ( ( $user->isLoggedIn() and $validCategory ) or $whitePage ) {
                return true;
        }
        return false;
}
```

Edit Localsettings.php

Activate the extension

Add the following to your LocalSettings.php

```
require_once
"$IP/extensions/RestrictAccessByCategoryAndGroup/RestrictAccessByCategoryAndG
roup.php";
```

Add the security groups

Create the groups as needed like the example here below. Make sure you keep track of your security and access matrix, this can very rapidly become extremely complex.

```
$wgGroupPermissions['Financial no public data']['*'] = true;
$wgGroupPermissions['Financial private data']['private'] = true;
```

Back to top of page - Back to Welcome Page