

TYPO3 Core - Feature #95175

security.ifByRoles-Viewhelper, which allows to check different groups and allows mixed use of usergroup-IDs and -names

2021-09-10 10:11 - Dieter Porth

Status:	New	Start date:	2021-09-10
Priority:	Should have	Due date:	
Assignee:		% Done:	0%
Category:	Fluid	Estimated time:	0.00 hour
Target version:	Candidate for patchlevel	Complexity:	
PHP Version:	7.3	Sprint Focus:	
Tags:			

Description

Let the code speak.

The ifHasRole-Viewhelper is too limited.

```
namespace TYPO3\CMS\Fluid\ViewHelpers\Security;
```

```
/*
 * This file is part of the TYPO3 CMS project.
 *
 * It is free software; you can redistribute it and/or modify it under
 * the terms of the GNU General Public License, either version 2
 * of the License, or any later version.
 *
 * For the full copyright and license information, please read the
 * LICENSE.txt file that was distributed with this source code.
 *
 * The TYPO3 project - inspiring people to share!
 */
```

```
use TYPO3\CMS\Core\Context\Context;
use TYPO3\CMS\Core\Context\UserAspect;
use TYPO3\CMS\Core\Utility\GeneralUtility;
use TYPO3Fluid\Fluid\Core\ViewHelper\AbstractConditionViewHelper;
```

```
/*
 * This ViewHelper implements an ifHasRole/else condition for frontend groups.
 *
 * Examples
 * =====
 *
 * Basic usage
 * -----
 *
 * ::
 *   normal-case
 *
 *   <f:security.switchByRoles roles=" Administrator, 5 " >
 *     <f:then> USER contain to at least one role. </f:then>
 *     <f:else> USER IS UNLOGGED or USER contain none of the usergroups. , The default-limiter
 is `.`.</f:else>
 *   </f:security.switchByRoles>
 *
 *
 *   anti-normal-case
 *
 *   <f:security.switchByRoles roles="Administrator;5" limiter=";" unlogged="true">
 *     <f:then> USER IS UNLOGGED or USER contain to at least one of the roles. The limiter is
 `.`.</f:then>
```

```

*         <f:else> USER contain none of the usergroups</f:else>
*     </f:security.switchByRoles>
*
*
*/
class IfByRolesViewHelper extends AbstractConditionViewHelper
{
    protected const BRANCH_THEN = true;
    protected const BRANCH_ELSE = false;
    protected const ATTR_ROLES = 'roles';
    protected const ATTR_LIMITER = 'limiter';
    protected const ATTR_UNLOGGED = 'unlogged';

    /** Initializes the "roles", "limiter" and "unlogged"-flag as argument.
    */
    public function initializeArguments()
    {
        parent::initializeArguments();
        $this->registerArgument(self::ATTR_ROLES, 'string', 'Whitespaces will be trimmed. A limited (comma) separated list of usergroups, which are represented by name or id. If the user contain one of the groups, the THEN-branch will shown. ');
        $this->registerArgument(self::ATTR_LIMITER, 'string', 'The limiter for your list.', false, ',');
        $this->registerArgument(self::ATTR_UNLOGGED, 'boolean', 'If `false` is set, the ELSE-branch will shown to an unlogged user and the user defined by roles will view the THEN-branch. If `true` is set, the THEN-branch will shown to an unlogged user and the THEN-branch will shown to user, which contain at least one of the roles. ', false, false);
    }

    /**
     * This method decides if the condition is TRUE or FALSE. It can be overridden in extending viewhelpers to adjust functionality.
     *
     * @param array $arguments ViewHelper arguments to evaluate the condition for this ViewHelper, allows for flexibility in overriding this method.
     * @return bool
     */
    protected static function evaluateCondition($arguments = null)
    {
        /** @var UserAspect $userAspect */
        $userAspect = GeneralUtility::makeInstance(Context::class)->getAspect('frontend.user');
        $roles = $arguments[self::ATTR_ROLES];
        if ((!$userAspect->isLoggedIn()) || (empty($roles)))
        {
            return (empty($arguments[self::ATTR_ROLES]) ? self::BRANCH_ELSE : self::BRANCH_THEN);
        }
        $limiter = ($arguments[self::ATTR_LIMITER] ?? ',');
        $roleList = array_filter(
            array_map(
                'trim',
                explode($limiter, $roles)
            )
        );
        $groupIds = $userAspect->getGroupIds();
        $getGroupNames = $userAspect->getGroupNames();
        $groups = array_merge($groupIds, $getGroupNames);
        if (!empty(array_intersect($groups, $roleList))) {
            return self::BRANCH_THEN;
        }
        return self::BRANCH_ELSE;
    }
}

```

Associated revisions

Revision c71d9a56 - 2021-09-17 15:29 - Benjamin Franzke

[BUGFIX] Avoid TSFE injection into standalone ContentObjectRenderer

TypoScriptFrontendController might not be available by the time ContentObjectRenderer instances are created

The constructor argument ContentObjectRenderer is intended to be used by TypoScriptFrontendController to inject itself, but not for injecting TypoScriptFrontendController when ContentObjectRenderer is created as a standalone instance.

This has already been adapted in the master branch by #95174 and is therefore backported as a separate 10.4-only fix.

Resolves: #95118

Related: #91625

Related: #95175

Releases: 10.4

Change-Id: Ie7f60cb40e78ec706019781ccd514acc080eb94a

Reviewed-on: <https://review.typo3.org/c/Packages/TYPO3.CMS/+71120>

Tested-by: Simon Gilli <typo3@gilbertsoft.org>

Tested-by: core-ci <typo3@b13.com>

Tested-by: Torben Hansen <derhansen@gmail.com>

Tested-by: Benjamin Franzke <bfr@qbus.de>

Reviewed-by: Simon Gilli <typo3@gilbertsoft.org>

Reviewed-by: Torben Hansen <derhansen@gmail.com>

Reviewed-by: Benjamin Franzke <bfr@qbus.de>

History

#1 - 2021-10-07 08:52 - Benni Mack

- Target version changed from 11 LTS to Candidate for patchlevel