TYPO3.Fluid - Feature # 41439

Status:	Rejected	**Priority:**	Should have
Author:	Michael Gerdemann	**Category:**	ViewHelpers
Created:	2012-09-28	**Assigned To:**	Bastian Waidelich
Updated:	2012-12-12	**Due date:**	
Has patch:	No		
Subject:	Argument to add an active-state class to "f:link.action"		

Description

If the rendered template exactly matches the action, it would be great if you would have the ability to automatically add a "active" class.

History

#1 - 2012-09-28 10:30 - Michael Gerdemann

Here is my edited render class:

```
/**
   * Render the link.
   *
   * @param string $action Target action
   * @param array $arguments Arguments
   * @param string $controller Target controller. If NULL current controllerName is used
   * @param string $package Target package. if NULL current package is used
   * @param string $subpackage Target subpackage. if NULL current subpackage is used
   * @param string $section The anchor to be added to the URI
   * @param string $format The requested format, e.g. ".html"
   * @param string $addActiveClass Add an active-class
   * @param array $additionalParams additional query parameters that won't be prefixed like $arguments (overrule $arguments)
   * @param boolean $addQueryString If set, the current query parameters will be kept in the URI
   * @param array $argumentsToBeExcludedFromQueryString arguments to be removed from the URI. Only active if $addQueryString =
TRUE
   * @return string The rendered link
   * @throws \TYPO3\Fluid\Core\ViewHelper\Exception
   * @api
   */
  public function render($action = NULL, $arguments = array(), $controller = NULL, $package = NULL, $subpackage = NULL, $section = '',
$format = '', $addActiveClass = '', array $additionalParams = array(), $addQueryString = FALSE, array
$argumentsToBeExcludedFromQueryString = array()) {
    $uriBuilder = $this->controllerContext->getUriBuilder();

    try {
      $uri = $uriBuilder
        ->reset()
        ->setSection($section)
        ->setCreateAbsoluteUri(TRUE)
        ->setArguments($additionalParams)
        ->setAddQueryString($addQueryString)
        ->setArgumentsToBeExcludedFromQueryString($argumentsToBeExcludedFromQueryString)
        ->setFormat($format)
```

```
                ->uriFor($action, $arguments, $controller, $package, $subpackage);
            $this->tag->addAttribute('href', $uri);
        } catch (\TYPO3\FLOW3\Exception $exception) {
            throw new \TYPO3\Fluid\Core\ViewHelper\Exception($exception->getMessage(), $exception->getCode(), $exception);
        }

        if(
            (
                !isset( $action )
                ||
                ( isset( $action ) && $this->controllerContext->getRequest()->getControllerActionName () == $action )
            )
            &&
            (
                !isset( $controller )
                ||
                ( isset( $controller ) && $this->controllerContext->getRequest()->getControllerName() == $controller )
            )
            &&
            (
                !isset( $package )
                ||
                ( isset( $package ) && $this->controllerContext->getRequest()->getControllerPackageKey () == $package )
            )
            &&
            (
                !isset( $subpackage )
                ||
                ( isset( $subpackage ) && $this->controllerContext->getRequest()->getControllerSubpackageKey () == $subpackage )
            )
        ) {
            $classTag = $this->tag->getAttribute('class');
            $this->tag->addAttribute('class', ( isset($classTag) && !empty( $classTag ) ? $classTag . ' ' : '' ) . $addActiveClass );
        }

        $this->tag->setContent($this->renderChildren());
        $this->tag->forceClosingTag(TRUE);

        return $this->tag->render();
    }
```

#2 - 2012-11-23 12:07 - Bastian Waidelich

- Status changed from New to Rejected

- Assigned To set to Bastian Waidelich

As discussed on IRC: This behavior is too specific to be implemented into a core ViewHelper in my opinion. Besides it probably won't work reliably in nested requests (plugins, widgets, ...).
Therefore I suggested to either subclass the link.action ViewHelper and adjust it as needed:

```
namespace Your\Package\ViewHelpers\Link;

class ActionViewHelper extends \TYPO3\Fluid\ViewHelpers\Link\ActionViewHelper {

    public function render($action = NULL, $arguments = array(), $controller = NULL, $package = NULL, $subpackage = NULL, $section = '', $format = '',  array $additionalParams = array(), $addQueryString = FALSE, array $argumentsToBeExcludedFromQueryString = array()) {

        if ($this->matchesCurrentRequest($action, $arguments, $controller, $package, $subpackage)) {
            $cssClass = 'active';
            if ($this->hasArgument('class')) {
                $cssClass .= ' ' . $this->arguments['class'];
            }
            $this->tag->addAttribute('class', $cssClass);
        }

        return call_user_func_array('parent::render', func_get_args());
    }

    protected function matchesCurrentRequest($action, $arguments, $controller, $package, $subpackage) {
        // TODO compare arguments with $this->controllerContext->getRequest() and return TRUE if they match
    }

}
```

Alternatively, you could create a subclass of the AbstractConditionViewHelper (see if ViewHelper for an example) that compares its arguments to the current request.
This one could be used as follows:

```
<f:link.action action="foo" class="{x:ifCurrentRequest(action: 'foo', then: ' active')}">Link</f:link.action>
```

#3 - 2012-12-12 14:36 - Bastian Waidelich

Bastian Waidelich wrote:

BTW: If you want to render a menu with active/inactive menu items this snippet might help you: https://gist.github.com/4223855