

## TYPO3.Flow - Feature # 55870

<b>Status:</b>	New	<b>Priority:</b>	Must have
<b>Author:</b>	Kay Strobach	<b>Category:</b>	
<b>Created:</b>	2014-02-11	<b>Assigned To:</b>	Christian Müller
<b>Updated:</b>	2014-04-15	<b>Due date:</b>	
<b>PHP Version:</b>	5.4		
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	Enhance f:form.textfield or add a f:form.datefield VH with enhanced validation and propertymapping		
<b>Description</b>			
currently datetime field handling is a bit of a mess.			
The idea is to have a viewHelper, which is used like this:			
<pre>&lt;f:form.datefield property="datetimeAttribute" format="d.m.Y" /&gt;</pre>			
Jobs of the VH:			
<ol style="list-style-type: none"><li>1. DateTime object is formatted as d.m.Y</li><li>2. Property mapping knows the format of the field it's d.m.Y and can automatically cast the passed string to a DateTime object</li></ol>			
The VH can create a hidden field with the datetime format and a visible date field, then the DateTimeConverter can use this information to successfully cast the object to a			

### History

#### #1 - 2014-02-11 11:22 - Kay Strobach

note conversion is done in DateTimeConverter

#### #2 - 2014-03-24 18:37 - Kay Strobach

The idea is to have a ViewHelper, which adds an additional hidden signed field which contains the valid date format (e.g. d.m.Y) after sending the form the string is then checked against the formatstring which was set for the field

#### #3 - 2014-04-09 11:41 - Kay Strobach

thanks to Christian Müller (thumbsup) i got the relevant hint on how to do that:

```
<?php

namespace TYPO3\FLOW\ViewHelpers\Form;

use TYPO3\Fluid\Core\ViewHelper\Exception;

class DateTimeTextfieldViewHelper extends \TYPO3\Fluid\ViewHelpers\Form\AbstractFormFieldViewHelper {
    /**
```

```

* @var string
*/
protected $tagName = 'input';

/**
 * Initialize the arguments.
 *
 * @return void
 * @api
 */
public function initializeArguments() {
    parent::initializeArguments();
    $this->registerTagAttribute('disabled', 'string', 'Specifies that the input element should be disabled when the page loads');
    $this->registerTagAttribute('maxlength', 'int', 'The maxlength attribute of the input field (will not be validated)');
    $this->registerTagAttribute('readonly', 'string', 'The readonly attribute of the input field');
    $this->registerTagAttribute('size', 'int', 'The size of the input field');
    $this->registerTagAttribute('placeholder', 'string', 'Placeholder for the field');
    $this->registerArgument('errorClass', 'string', 'CSS class to set if there are errors for this view helper', FALSE, 'f3-form-error');
    $this->registerArgument('dateFormat', 'string', 'DateTime object attribute', FALSE, 'Y-m-d\TH:i:sP');
    $this->registerUniversalTagAttributes();
}

public function render($required = NULL, $type = 'text') {
    $name = $this->getName();
    $this->registerFieldNameForFormTokenGeneration($name);

    $this->tag->addAttribute('type', $type);
    $this->tag->addAttribute('name', $name . '[date]');

    /** @var \DateTime $value */
    $value = $this->getValue(FALSE);
    if(is_a($value, '\DateTime')) {
        $valueString = $value->format($this->arguments['dateFormat']);
    } else {
        throw new Exception('Expected \DateTime object, but got ' . gettype($value) . ' for ' . $this->getName());
    }

    if ($value !== NULL) {
        $this->tag->addAttribute('value', $valueString);
    }

    if ($required !== NULL) {
        $this->tag->addAttribute('required', 'required');
    }

    $this->setErrorClassAttribute();

    return $this->tag->render() . '<input type="hidden" value="" . htmlspecialchars($this->arguments['dateFormat']) . "" name="" .
    $this->getName() . '[dateFormat]' />';
}
}

```

**#4 - 2014-04-15 12:32 - Christian Müller**

- *Assigned To set to Christian Müller*