

TYPO3.Fluid - Bug # 50555

Status:	Closed	Priority:	Should have
Author:	Thomas Christiansen	Category:	
Created:	2013-07-30	Assigned To:	Bastian Waidelich
Updated:	2013-07-31	Due date:	
Has patch:	No		
Affected Flow version:	(any)		
Subject:	Can't get an empty array in if condition		
Description			
<p>Hi all,</p> <p>i am having trouble to get an empty array in a shorthand if condition.</p> <p>The situation is the following: I have an Input field which can be required or optional, depending on a checkbox in Backend.</p> <p>In my Template i tried this;</p> <pre>additionalAttributes="{f:if(condition: '{field.mandatory}', then: '{required: 1}', else: '{}')}"</pre> <p>The "then"-case works fine, i am getting the attribute required="1" as expected.</p> <p>The problem is, that the '{}' in the else-case returns an empty string, not an empty array. Then i get an exception, because additionalAttributes expects an array.</p> <p>I tried a couple of things to get this to work like {0: null}, {\''} or {null} but i always get either an empty string or a non-empty array like array(0 => null).</p> <p>The last one is not an option because it would result in an attribute "0" in the html tag.</p> <p>Me and my colleagues came up with a view workarounds:</p> <ol style="list-style-type: none">1) Assign an empty array to a variable in the controller and use this variable in the template. - This is in my case not an option because i am extending an TER extension and dont want to override its controller.2) Add an "Empty-Array-Getter" to the Model like getEmptyArray which returns an empty array. - This is a possibility, because i am extending the model as well, but... urghs ;)3) Write a viewhelper that adds an empty array to the templateVariableContainer. - This is probably the best workaround and i am using this for now. <p>The viewhelper looks simply like this:</p> <pre>class EmptyArrayViewHelper extends \TYPO3\CMS\Fuild\Core\ViewHelper\AbstractViewHelper { public function render() { \$this->templateVariableContainer->add('emptyArray', array()); return NULL; } }</pre> <p>So i use can {emptyArray} in my template now after calling the viewhelper:</p> <pre><fx:EmptyArray /> ...</pre>			

```
additionalAttributes="{f:if(condition: '{field.mandatory}', then: '{required: 1}', else: '{emptyArray}')}"
```

Did we miss anything here or is there no other way to solve this scenario?

Thanks and regards,
Thomas

History

#1 - 2013-07-30 13:11 - Bastian Waidelich

- Status changed from New to Needs Feedback
- Assigned To set to Bastian Waidelich

What about

```
1 <f:if condition="{field.mandatory}">
2 <f:then>
3 <f:form.checkbox property="foo" additionalAttributes="{required: true}" />
4 </f:then>
5 <f:else>
6 <f:form.checkbox property="foo" />
7 </f:else>
8</f:if>
```

If you need it more than once you could put that code into a partial and use it like this:

```
1 <f:render partial="checkbox" arguments="{requiredCondition: field.mandatory}" />
```

(and then change **{field.mandatory}** to **{requiredCondition}** in the partial)

#2 - 2013-07-31 15:08 - Thomas Christiansen

Yes sure, that is another workaround.

Actually this is where I started and what i did not want because its annoying to maintain.

Thanks for your reply but i am going with the viewhelper solution.

Regards,
Thomas

#3 - 2013-07-31 15:21 - Bastian Waidelich

- Status changed from Needs Feedback to Closed

Thomas Christiansen wrote:

| *Thanks for your reply but i am going with the viewhelper solution.*

Ok, I'm closing the issue for now because that is actually a missing feature that would be quite hard to implement without huge benefit.

Some background information:

The Fluid argument notation is a fully compatible subset of the JavaScript Object Notation (JSON). For an empty array we'd need to add support for a new symbol:

[]

This would increase complexity and create conflicts with existing Fluid templates.

Feel free to add comments or re-open if you don't agree