TYPO3.Fluid - Bug # 33551

Status:	New	Priority:	Must have
Author:	Stephan Schuler	Category:	Core
Created:	2012-01-31	Assigned To:	Sebastian Kurfuerst
Updated:	2013-06-05	Due date:	

Has patch:	No
Affected Flow version:	(any)

Subject:	View helper values break out of a partial scope

Description

If a partial is called at least twice which has a certain view helper inside, the last occurence of the inside value breaks out of the partials scope and will be the result of all usages of this view helper from now on.

This only appears when the first call triggers the caching mechanism but executes the uncached template (AbstractTemplateView::render() $parsedTemplate is instance of "TYPO3\Fluid\Core\Parser\ParsingState").
As soon as the template is a compiled one (AbstractTemplateView::render() $parsedTemplate is instance of "TYPO3\Fluid\Core\Compiler\AbstractCompiledTemplate"), the result is as expected.

Main template:

```
<html>
  <head>
    <title>Little bug</title>
  </head>
  <body>
    <ol>
      <li><f:escape type="xml">I am pre partial</f:escape></li>
      <f:render partial="Demo" arguments="{value: 'I am the first node within a partial'}" />
      <f:render partial="Demo" arguments="{value: 'I am not the first but the second node within a partial'}" />
      <li><f:escape type="xml">I am post partial</f:escape></li>
    </ol>
  </body>
</html>
```

Demo-Partial:

```
<li><f:escape type="xml">{value}</f:escape></li>
```

The expected result would be:

1. I am pre partial
2. I am the first node within a partial
3. I am not the first but the second node within a partial
4. I am post partial

But the given result is:

1. I am pre partial
2. I am the first node within a partial
3. I am not the first but the second node within a partial
4. I am not the first but the second node within a partial

Related issues:

related to TYPO3.Fluid - Bug # 47674: HtmlentitiesViewHelper is scope singleton	**Resolved**	**2013-04-28**

History

#1 - 2012-03-08 20:52 - Bastian Waidelich

- Category set to Core

- Status changed from New to Accepted

Wow, thanks for the detailed description

#2 - 2012-03-08 21:23 - Bastian Waidelich

- Status changed from Accepted to Needs Feedback

- Assigned To set to Bastian Waidelich

I fail to reproduce this with a slightly modified example (as the escape viewhelper is not longer part of the distribution):

.h3 Templates/Index.html

```
 1<html>
 2<head>
 3   <title>Little bug</title>
 4</head>
 5<body>
 6<ol>
 7   <li><f:format.htmlspecialchars>I am pre partial</f:format.htmlspecialchars></li>
 8   <f:render partial="test" arguments="{value: 'I am the first node within a partial'}" />
 9   <f:render partial="test" arguments="{value: 'I am not the first but the second node within a partial'}" />
10   <li><f:format.htmlspecialchars>I am post partial</f:format.htmlspecialchars></li>
11</ol>
12</body>
13</html>
```

.h3 Partials/test.html

```
1<li><f:format.htmlspecialchars>{value}</f:format.htmlspecialchars></li>
```

Can you please verify this again?

#3 - 2012-03-21 14:44 - Bastian Waidelich

Bastian Waidelich wrote:

> *I fail to reproduce this [...]*

@Stephan can you please confirm this, otherwise I'll close the issue for now

#4 - 2012-10-25 13:58 - Bastian Waidelich

- Status changed from Needs Feedback to Closed

- Affected Flow version changed from Git master to (any)

Closed because of missing feedback

#5 - 2013-01-18 14:31 - Stephan Schuler

- Status changed from Closed to New

He Bastian.

Sorry for not responding, but forge didn't notify me and then I somehow went over this.

Unfortunatelly the error still persist.

The Fluid code looks like this:

```
1<?xml version="1.0" encoding="UTF-8"?>
2<rendertest>
3    <line><f:format.htmlentities>first main</f:format.htmlentities></line>
4    <f:render partial="Partialtest" arguments="{first: 'first partial, first line', second: 'first partial, second line'}" />
5    <f:render partial="Partialtest" arguments="{first: 'second partial, first line', second: 'second partial, second line'}" />
6    <line><f:format.htmlentities>last main</f:format.htmlentities></line>
7</rendertest>
8
```

```
1<?xml version="1.0" encoding="UTF-8"?>
2<rendertest>
3    <line><f:format.htmlentities>first main</f:format.htmlentities></line>
4    <f:render partial="Partialtest" arguments="{first: 'first partial, first line', second: 'first partial, second line'}" />
5    <f:render partial="Partialtest" arguments="{first: 'second partial, first line', second: 'second partial, second line'}" />
6    <line><f:format.htmlentities>last main</f:format.htmlentities></line>
7</rendertest>
```

Output of the first request (here the template is not cached, gets rendered and then pushed to the caching directory):

```
1<?xml version="1.0" encoding="UTF-8"?>
2<rendertest>
```

```
 3    <line>first main</line>
 4    <line>
 5    first partial, first line
 6    first partial, second line
 7</line>
 8    <line>
 9    second partial, first line
10    second partial, second line
11</line>
12    <line>second partial, second line</line>
13</rendertest>
```

Output of all following requests (here the template has a representation in the Fluid caching directory and so gets only executed):

```
 1<?xml version="1.0" encoding="UTF-8"?>
 2<rendertest>
 3    <line>first main</line>
 4    <line>
 5    first partial, first line
 6    first partial, second line
 7</line>
 8    <line>
 9    second partial, first line
10    second partial, second line
11</line>
12    <line>last main</line>
13</rendertest>
```

So, nothing changed here.

Changing from <f:format.htmlentities>{foo}</f:format.htmletities> to {foo -> f:format.htmlentities} doesn't change anything.

#6 - 2013-01-21 22:15 - Sebastian Kurfuerst

That's very weird behavior. I'd try to reproduce it inside the viewhelpertest packages, I just cannot promise when I'll find the time.

But that looks like a bug worth fixing ;)

#7 - 2013-04-28 12:19 - Alexander Berl

Some findings:

I think the problem comes from how ParsingState, alas TemplateParser initializes singleton view helpers vs. how AbstractCompiledTemplate does. If you look closely, the format.htmlentities ViewHelper is of scope singleton, while the format.htmlspecialchars is not - that would explain why Bastian couldn't reproduce the problem.
On a closer look, AbstractCompiledTemplate calls viewHelper::resetState on singletons, while TemplateParser doesn't.

After a fix it should be reviewed if htmlentities needs to stay singleton, also the htmlspecialchars ViewHelper contains special code for the ::compile

function, while none of the other format viewhelpers does. This cannot be correct to my understandings.

#8 - 2013-04-28 20:40 - Alexander Berl

I reproduced the bug with the Viewhelpertests, unfortunately the problem isn't only the missing resetState() as I suspected.

I also tried simply working around the problem by always using the compiled template even on first load in AbstractTemplateView, but that still doesn't solve the problem oddly enough:

```
public function render($actionName = NULL) {
    $this->baseRenderingContext->setControllerContext($this->controllerContext);
    $this->templateParser->setConfiguration($this->buildParserConfiguration());

    $templateIdentifier = $this->getTemplateIdentifier($actionName);
    if ($this->templateCompiler->has($templateIdentifier)) {
        $parsedTemplate = $this->templateCompiler->get($templateIdentifier);
    } else {
        $parsedTemplate = $this->templateParser->parse($this->getTemplateSource($actionName));
        if ($parsedTemplate->isCompilable()) {
            $this->templateCompiler->store($templateIdentifier, $parsedTemplate);
                // Directly use the compiled template
            $parsedTemplate = $this->templateCompiler->get($templateIdentifier);
        }
    }
    ...
}
```


I hence suspect that there is something other wrong with the context or alike, which I can't really dig into atm.

I pushed my Viewhelpertests case at https://review.typo3.org/20270, so Bastian and/or Sebastian can work from there.

I will however add a new issue that just will remove the singleton scope from the htmlentities viewhelper, which will at least be a cosmetic fix for this issue.

#9 - 2013-05-31 18:14 - Bastian Waidelich

- Assigned To changed from Bastian Waidelich to Sebastian Kurfuerst

Alexander Berl wrote:

> *I reproduced the bug with the Viewhelpertests, unfortunately the problem isn't only the missing resetState() as I suspected.*

Thanks a lot for preparing this so nicely, Alexander!

@Sebastian: I'm assigning this to you only for feedback

I finally came around having a closer look at this again and here's my finding:

If I remove the @Flow\Scope("singleton") from the HtmlentitiesViewHelper your second test case still fails **during parsing** on the second hit everything works.

If I instead remove lines 41 - 46 of the AbstractCompiledTemplate, everything works as expected..

Here my suggestion:
First of all we should get rid of the singleton scope.. HtmlentitiesViewHelper is the only VH that is a singleton – and I can't exactly remember why, probably for performance reasons (this VH is used a lot because it's added automatically by Fluid for HTML templates).
To save memory and increase performance we should implement the **CompilableInterface** instead.
This looks like that https://gist.github.com/bwaidelich/e0e34ba95fb2b2306dab and fixes the issue for me without further adjustments – we should do that for other core VHs to IMO

Anyways, I still didn't completely get why the output of the HtmlentitiesViewHelper is "cached" otherwise..

#10 - 2013-05-31 18:18 - Bastian Waidelich

Bastian Waidelich wrote:

A little addition:

> If I remove the @Flow\Scope("singleton") from the HtmlentitiesViewHelper your second test case still fails **during parsing** on the second hit
> everything works.

Obviously, because your second test uses a new ViewHelper "Singleton" that is.. of scope singleton. If I remove that too, everything works too

#11 - 2013-06-05 14:41 - Alexander Berl

Bastian Waidelich wrote:

> Bastian Waidelich wrote:
>
> A little addition:
>
> > If I remove the @Flow\Scope("singleton") from the HtmlentitiesViewHelper your second test case still fails **during parsing** on the second hit
> > everything works.
>
> Obviously, because your second test uses a new ViewHelper "Singleton" that is.. of scope singleton. If I remove that too, everything works too

Yes, I added that so there's still a test for the singleton-misbehavior even when the htmlentities viewhelper is not singleton any more :)

> To save memory and increase performance we should implement the CompilableInterface instead.

Is there any profiling-suite or setup you use for verifying the performance/memory influence of changes?