

TYPO3.Fluid - Bug # 8879

| | | | |
|--|------------------------------|---------------------|-------------------|
| Status: | Closed | Priority: | Should have |
| Author: | Adrian Föder | Category: | |
| Created: | 2010-07-15 | Assigned To: | |
| Updated: | 2013-12-04 | Due date: | |
| Has patch: | No | | |
| Affected Flow version: | (any) | | |
| Subject: | { } when using as JavaScript | | |
| Description | | | |
| assuming the following FLUID code: | | | |
| <pre><f:if condition="foo"> <script type="text/javascript"> var myDemoJSObject = { key1: 'foo', key2: 'bar' } </script> </f:if></pre> | | | |
| this is outputted as | | | |
| <pre><script type="text/javascript"> var myDemoJSObject = Array </script></pre> | | | |
| I think the brackets are regarded as inline view helpers and tried to be parsed. Wouldn't it be senseful to require inline view helpers may NOT have a whitespace after the opening bracket, so it should be (regex) /\{w/ and NOT /\{./ or something like that. | | | |
| Related issues: | | | |
| related to Fluid - ViewHelper - Incubator - Suggestion # 3677: embed.javascript | | New | 2009-06-16 |
| related to TYPO3.Fluid - Bug # 44140: Unlucky JavaScript syntax can break inl... | | Resolved | 2012-12-20 |
| related to TYPO3.Fluid - Feature # 39682: Disregegard { as Fluid variable whe... | | Rejected | 2012-08-10 |
| duplicates TYPO3.Fluid - Bug # 6338: JSON is being parsed as string Array | | Closed | 2010-02-04 |

History

#1 - 2010-07-15 14:06 - Bastian Waidelich

- Status changed from New to Closed

Hi Adrian, thanks for your feedback.

This seems to be the same issue as #6338.

Because it would be really difficult to differenciate Fluid brackets from JavaScript brackets automatically you should either extract your script to a js file or wrap your inline scripts like:

```
<script type="text/javascript">*<![CDATA[*]
  // your scripts
/*]]>*/</script>
```

That's considered best practice anyways.

have a look at #3677 too.

#2 - 2010-07-15 14:24 - Adrian Förder

Hi Bastian,

sorry for that double posting; I didn't find an already existing report similar to mine.

In fact, moving to an external js file is exactly what I would not want, because I have to use ViewHelpers inside my JavaScript:

```
<script type="text/javascript">
var myTranslations = {
  llKeyOne: '{f:translate(key:"myKey")}'
};
</script>
```

This was the thing that triggered my reporting.

But of course JavaScript content should be wrapped in CDATA anyway.

Am I understanding right that inline view helper parsing would NOT take place inside a CDATA section? If so, my approach above to get the translation into javascript would also fail....

So long and best regards.

Adrian

#3 - 2010-07-15 15:24 - Bastian Waidelich

Adrian Foeder wrote:

Hi Adrian,

Am I understanding right that inline view helper parsing would NOT take place inside a CDATA section?

Exactly. CDATA is the proper (and only) way to comment out Fluids inline notation. So it's kind of similar to the `<?php ?>` tags in PHP. If you want to mix JavaScript and Fluid, you'll could write

```
<script type="text/javascript">/*!CDATA[*/
/*]]>*/
var llKeyOne = '{f:translate(key:"key1")}';
var llKeyTwo = '{f:translate(key:"key2")}';
/*!CDATA[*/
var myTranslations = {
  llKeyOne: llKeyOne,
```

```
llKeyTwo: llKeyTwo
};
/*]]>*/</script>
```

HTH,
Bastian

#4 - 2010-07-15 15:44 - Adrian Förder

Hi Bastian,

but why, and sorry for the question, not make it mandatory to let the namespace identifier immediately follow after the opening bracket:

"valid" inline ViewHelper would be

```
{f:viewHelperName(attrib:'etc')}
```

while the "invalid" way were

```
{ f:viewHelperName(attrib:'etc')}
```

An opening bracket then is only regarded as inline View Helper when having no white space between the opening bracket and "anything else", or vice versa: if a whitespace (or tab or enter or a char not being alphanumeric) follows the opening bracket it is NOT regarded as inline view helper and therefore not be parsed?

I think such a rule could be legitime (MySQL functions for example must also not contain whitespaces before the opening bracket) and solve the "Javascript" problem...

#5 - 2013-07-17 12:05 - Fedir RYKHTIK

Quite annoying to put the CDATA before and after each JS array. Prefer to have some syntax optimization in Fluid, as {\$ or obligatory name space after the bracket opening.

#6 - 2013-07-17 12:11 - Bastian Waidelich

- Has patch set to No

- Affected Flow version set to (any)

Fedir RYKHTIK wrote:

Quite annoying to put the CDATA before and after each JS array. Prefer to have some syntax optimization in Fluid, as {\$ or obligatory name space after the bracket opening.

Do you still have issues with this since #43356 is merged??

Apart from that: You should really outsource **all** JavaScript to js files. Latest when [Content Security Policy](#) is going to be widely used, inline JS won't work any longer

#7 - 2013-12-04 13:01 - Fedir RYKHTIK

Bastian Waidelich wrote:

Do you still have issues with this since #43356 is merged??

*Apart from that: You should really outsource **all** JavaScript to js files. Latest when [Content Security Policy](#) is going to be widely used, inline JS won't work any longer*

I didn't noticed changes in current LTS version.

Thanks for the link, it's a good reason not to use Inline JS. I'm generating external JS files with Fluid, it's not inline, even if it's Fluid based.