

TYPO3.Neos - Task # 53046

Status:	Resolved	Priority:	Should have
Author:	Bastian Waidelich	Category:	Services
Created:	2013-10-22	Assigned To:	Bastian Waidelich
Updated:	2013-12-04	Due date:	

Subject: Reduce generated code of ContentElementWrappingService

Description

Rendering content elements currently creates a whole lot of attributes and additional wrapping elements. Especially the latter is an issue because it has an effect on the rendering.

Example of the rendering of a simple headline:

Current output in "live" workspace:

```
1 <div class="neos-contentcollection">
2   <div class="neos-contentelement typo3-neos-nodetypes-headline" id="c8f734d31-bd9f-4487-bf4d-033ea5c52a4d">
3     <div>
4       <h1>Some headline</h1>
5     </div>
6   </div>
7 </div>
```

Expected output in "live" workspace:

```
1 <h1>Some headline</h1>
```

Current output in "editor" workspace:

```
1 <div class="neos-contentcollection" about="/sites/neosdemotypo3org/home/main@user-admin" typeof="
typo3:TYPO3.Neos:ContentCollection" rel="typo3:content-collection" data-neos-_typoscript-path="page&lt;TYPO3.Neos:Page
&gt;/body&lt;TYPO3.TypoScript:Template&gt;/content/main&lt;TYPO3.Neos:ContentCollection&gt;"
data-neos-__workspacename="user-admin">
2   <div class="neos-contentelement typo3-neos-nodetypes-headline" id="c8f734d31-bd9f-4487-bf4d-033ea5c52a4d" typeof="
typo3:TYPO3.Neos.NodeTypes:Headline" about="/sites/neosdemotypo3org/home/main/headline1@user-admin" tabindex="0"
data-neos-__workspacename="user-admin" data-neos-_typoscript-path="page&lt;TYPO3.Neos:Page&gt;/body&lt;
TYPO3.TypoScript:Template&gt;/content/main&lt;TYPO3.Neos:ContentCollection&gt;/itemRenderer&lt;
TYPO3.Neos:ContentCase&gt;/default&lt;TYPO3.TypoScript:Matcher&gt;/element&lt;TYPO3.Neos.NodeTypes:Headline&gt;"
data-neos-_removed="false" data-neosdatatype-_removed="xsd:boolean" data-neos-_hidden="false"
data-neosdatatype-_hidden="xsd:boolean" data-neos-_hidden-before-date-time=""
data-neosdatatype-_hidden-before-date-time="xsd:date" data-neos-_hidden-after-date-time=""
data-neosdatatype-_hidden-after-date-time="xsd:date" data-neos-title="&lt;h1&gt;I'm here to be defaced ;-)&lt;/h1&gt;"
data-neos-__nodetype="TYPO3.Neos.NodeTypes:Headline">
3     <div property="typo3:title" class="neos-inline-editable"><h1>Some headline</h1></div>
4   </div>
5 </div>
```

Expected output in "editor" workspace (exemplary):

```
1 <neos-contentcollection data-neos-node-type="typo3:TYPO3.Neos:ContentCollection" neos-node-path="
/sites/neosdemotypo3org/home/main" data-neos-typoscript-path="page&lt;TYPO3.Neos:Page&gt;/body&lt;
TYPO3.TypoScript:Template&gt;/content/main&lt;TYPO3.Neos:ContentCollection&gt;" data-neos-workspace-name="
user-admin">
2   <h1 data-neos-node-type="typo3:TYPO3.Neos.NodeTypes:Headline" neos-node-property="title" neos-node-path="
```

```
/sites/neosdemotypo3org/home/main/headline1" data-neos-typoscript-path="page&lt;TYPO3.Neos:Page&gt;/body&lt;TYPO3.TypoScript:Template&gt;/content/main&lt;TYPO3.Neos:ContentCollection&gt;/itemRenderer&lt;TYPO3.Neos:ContentCase&gt;/default&lt;TYPO3.TypoScript:Matcher&gt;/element&lt;TYPO3.Neos.NodeTypes:Headline&gt;" tabindex="0">Some headline</h1>
3</neos-contentcollection>
```

h2.General ideas (to be discussed!):

- Reduce nesting levels: If the content element contains a root element, modify it instead of adding an additional wrapping tag
- If we need to add a wrapping element, we could use non-existing elements prefixed with neos- (e.g. neos-contentelement). This is better to distinguish for the user and it is less immersive because browsers (should) ignore unknown elements.
- only use data-neos--prefixed attributes for consistency. They can be used in CSS selectors, too.
- No need to use underscores in attribute-names (e.g. data-neosdatatype_hidden as everything prefixed with neos- should be considered "private"
- Only add attributes for non-defaults (e.g. data-neos-removed="false" can be omitted)

Subtasks:

Task # 54016: Introduce Content TypoScript object

Resolved

Task # 54137: Deprecate ContentElement ViewHelper

Resolved

Associated revisions

Revision 3f69748e - 2013-11-11 12:37 - Christopher Hlubek

[FEATURE] Support tag name and attributes for ContentElementViewHelper

This change extends the neos:contentElement view helper that is used to wrap nodes with RDFa and editing metadata.

It adds a configurable tag name and registers universal tag attributes to override all attributes of the wrapping tag from the view helper.

This is the first step to reduce the generated code and nesting by allowing more fine-grained control over the markup.

Change-Id: Iadb881d9545d442b6816ec105c75ac62bd99cac2

Related: #53046

Reviewed-on: <https://review.typo3.org/25174>

Reviewed-by: Christopher Hlubek

Tested-by: Christopher Hlubek

Reviewed-by: Aske Ertmann

Tested-by: Mr. Hudson

Revision cf28fa90 - 2013-11-11 14:21 - Christopher Hlubek

[!!!][FEATURE] Support custom tag name and attributes for ContentCollection

This change adds new TypoScript properties to the ContentCollection object to set the tagName and attributes of the generated tag that wraps the collection.

Example::

```
prototype(Acme.Demo:List) < prototype(TYPO3.Neos:Template) {
    items = TYPO3.Neos:ContentCollection {
        nodePath = 'items'
```

```

    tagName = 'ul'
    attributes.class = 'my-list'
  }
}

```

The TypoScript code will render the following HTML::

```

<ul class="neos-contentcollection my-list">
...
</ul>

```

This is a breaking change as the "className" property for a custom class name on the generated tag is removed. It has to be changed to the "attributes.class" declaration.

Change-Id: Ia006e962eaaafb35d4e362900b093e969c9ebb9d

Related: #53046

Reviewed-on: <https://review.typo3.org/25179>

Tested-by: Christopher Hlubek

Tested-by: Mr. Hudson

Reviewed-by: Aske Ertmann

Reviewed-by: Rens Admiraal

Tested-by: Rens Admiraal

Reviewed-by: Christopher Hlubek

Revision e9415a10 - 2013-12-02 09:39 - Bastian Waidelich

[FEATURE] Improve contentElement.editable ViewHelper

This improves the EditableViewHelper so that it is easier to use and less error-prone.

It mainly makes specification of the property value optional.

Before::

```

<neos:contentElement.editable property="propertyName">
  {propertyName -> f:format.raw()}
</neos:contentElement>

```

Now::

```

{neos:contentElement.editable(property: 'propertyName')}

```

Note: This change is fully backwards compatible because any child nodes of this ViewHelper overrule the specified property argument.

It is however recommended not to set them as that can lead to unexpected results. It is also recommended to use Fluid's inline syntax for better readability and to make browsers render the placeholder when opening the template file directly.

Change-Id: I2e6e9035a85b58cd9f28f5f2f5b57b7631f64497

Related: #53046

Reviewed-on: <https://review.typo3.org/25765>

Reviewed-by: Bastian Waidelich

Tested-by: Bastian Waidelich

Reviewed-by: Christopher Hlubek

Tested-by: Christopher Hlubek

History

#1 - 2013-10-22 21:51 - Bastian Waidelich

A quick and dirty (but working) example on how to replace the wrapping tag by modified attributes of the root element:

```
1function wrapHtml($html, $node) {
2  $nodeAttributes = sprintf('data-neos-node-type="%s" data-neos-node-identifier="%s"', $node->type, $node->identifier);
3  $rootTagName = getHtmlRootTagName($html);
4  if ($rootTagName !== NULL) {
5    $modifiedHtml = preg_replace('/<(' . $rootTagName . ')/xi', '<$1 ' . $nodeAttributes, $html, 1, $numberOfReplacements);
6    if ($numberOfReplacements === 1) {
7      return $modifiedHtml;
8    }
9  }
10 return sprintf('<neos-contentelement %s>%s</neos-contentelement>', $nodeAttributes, $html);
11}
12
13function getHtmlRootTagName($html) {
14  $domDocument = new \DOMDocument();
15  // ignore parsing errors
16  libxml_use_internal_errors(TRUE);
17  $domDocument->loadHTML($html);
18  $xpath = new \DOMXPath($domDocument);
19  $rootElement = $xpath->query('/html/body/*');
20  if ($rootElement === FALSE || $rootElement->length !== 1) {
21    return NULL;
22  }
23  return $rootElement->item(0)->nodeName;
24}
25$node = new \stdClass();
26$node->type = 'test';
27$node->identifier = 'uuid';
```

```
1echo wrapHtml('<div class="foo"><p>lorem ipsum<br>dolor</p></div>', $node);
```

would produce:

```
1<div data-neos-node-type="test" data-neos-node-identifier="uuid" class="foo"><p>lorem ipsum<br>dolor</p></div>
```

And

```
1 echo wrapHtml('<div class="foo"><p>lorem ipsum<br>dolor</p></div><span>another root node</span>', $node);
```

would result in:

```
1 <neos-contentelement data-neos-node-type="test" data-neos-node-identifier="uuid">
2 <div class="foo"><p>lorem ipsum<br>dolor</p></div><span>another root node</span>
3 </neos-contentelement>
```

#2 - 2013-10-22 23:12 - Rens Admiraal

I fully agree on the cleaning up of the HTML, and as some of the changes could be breaking I would suggest doing it soon. On the other hand I still think we should stick to the guideline that we have the same HTML output in the frontend as in the backend by default. For this we will have to add wrapping code to make the backend work, but we should allow the integrator to suppress this. We should not render anything that's not absolutely required to meet that requirement.

One of the strong points of TYPO3 CMS is that you can alter absolutely everything regarding the output. We should offer the same in Neos.

Some comments on your points:

- No need to use underscores in attribute-names (e.g. data-neosdatatype-_hidden as everything prefixed with neos- should be considered "private")
The underscore for private is not used for making something private, but for internal properties which are stored directly on the node, without an underscore are normal node properties. I don't think we should touch that in this cleanup.
- Only add attributes for non-defaults (e.g. data-neos-removed="false" can be omitted)
If we do that we should make sure that the inspector will handle this correctly as the properties are also used to detect if a node has a modified state. Should be doable, just something to keep in mind. And I'm not sure if the gain is worth the effort.

And then about the attributes used: we need to stick to properties like property="typo3:title" as otherwise we break the RDFa approach which will break VIE which will disable all content editing ;-). We also still need the wrapping element around for example the <h1></h1> in a headline because otherwise an editor like aloha will not be able to change the blockformat.

#3 - 2013-10-23 11:00 - Bastian Waidelich

Rens Admiraal wrote:

Thanks for your feedback!

To avoid getting lost in discussions about (currently not so pressing) issues here again the actual suggestion I wanted to make with this ticket, but got lost in details (below replies to your feedback):

Currently the template for a simple NodeType looks like this:

```
1 {namespace neos=TYPO3\Neos\ViewHelpers}
2 <neos:contentElement node="{node}">
3 <div class="my-element">
4 <h2><neos:contentElement.editable property="title">{text -> f:format.raw()}</neos:contentElement.editable></h2>
5 <p><neos:contentElement.editable property="text">{text -> f:format.raw()}</neos:contentElement.editable></p>
```

```
6 </div>
7</neos:contentElement>
```

This is almost like writing binary code ;)

I would expect:

```
1{namespace neos=TYPO3\Neos\ViewHelpers}
2<div class="my-element">
3 <h2>{neos.contentElement.editable(property: 'title')}</h2>
4 <p>{neos.contentElement.editable(property: 'text')}</p>
5</div>
```

And the WrappingService (that knows all about the context and the current node) should take care of adding the required attributes in the backend (and forwarding the raw content in frontend).

In this case, because the rendered template contains a root element, it would add the attributes to this tag (I tested this with some freaky broken html and the fallback works reliably):

```
1<div class="my-element" data-neos-node-path="..." ..>
2 <h2>the title</h2>
3 <p>the text</p>
4</div>
```

Otherwise it would add a wrapping element:

```
1<neos-contentelement data-neos-node-path="..." ..>
2 ...
3</neos-contentelement>
```

Do you think it is worth working in that direction at all or do you think it is not possible/too much effort?

And for your other points:

| *One of the strong points of TYPO3 CMS is that you can alter absolutely everything regarding the output. We should offer the same in Neos.*

True, and the easiest way to achieve this, is (IMO) to not alter the output at all in live workspace (alter = adding tags/attributes on top of the rendered fluid templates) - if that's possible somehow.

| - *No need to use underscores in attribute-names (e.g. data-neosdatatype_hidden as everything prefixed with neos- should be considered "private"*

| *The underscore for private is not used for making something private, but for internal properties which are stored directly on the node, without an underscore are normal node properties. I don't think we should touch that in this cleanup.*

I know that those properties have an underscore in the node implementation to distinguish them from standard node properties. But I think the rendered HTML is a different context with a different model and I don't see a problem omitting the underscore there as we don't have the distinguish-issue there, or do we?

But I totally agree to you, we shouldn't touch this anytime soon - just a minor weirdness that I stumbled upon while looking at the generated source code. And I thought underscores would be invalid in attributes, but that's **not** the case for data attributes!

- Only add attributes for non-defaults (e.g. `data-neos-removed="false"` can be omitted)

If we do that we should make sure that the inspector will handle this correctly as the properties are also used to detect if a node has a modified state. Should be doable, just something to keep in mind. And I'm not sure if the gain is worth the effort.

Couldn't we just output those **when** the property is modified or != default - the node should know about this, or?

Well, I'm not sure about all the implications and complexity. I just collected issues that I found weird and I think that we should at least get rid of the (or some of the) redundant information at some point because it will bite us sooner or later (not only performance wise). For example:

```
1 <div
2   class="typo3-neos-nodetypes-headline"
3   typeof="typo3:TYPO3.Neos.NodeTypes:Headline"
4   data-neos-__nodetype="TYPO3.Neos.NodeTypes:Headline"
5
6   class="neos-contentelement-removed"
7   data-neos-_removed="true"
8
9   class="neos-contentelement-hidden"
10  data-neos-_hidden="true"
11
12 ... />
```

And then about the attributes used: we need to stick to properties like `property="typo3:title"` as otherwise we break the RDFa approach which will break VIE which will disable all content editing ;-)

Ok, I see. The "typo3:" prefix was just a complete blackbox to me. Shouldn't this be "neos:" btw?

We also still need the wrapping element around for example the `<h1></h1>` in a headline because otherwise an editor like aloha will not be able to change the blockformat.

That's a shame.. But in this case we should do an exception to the "frontend HTML = backend HTML" rule (I think) as the additional wrapping div is not acceptable for many. If we use an unknown element like `<neos-contenttype>` instead of a `<div>` in the backend, this should have no visual influence (even though it might break JS/CSS due to changed selectors of course, but then the integrator can add some CSS/JS snippets to make it work in backend, just like today).

#4 - 2013-11-13 17:16 - Bastian Waidelich

- Status changed from New to Accepted

- Assigned To set to Bastian Waidelich

I'll give this a try..

FYI: Update from yesterdays [team meeting](#) :

1) Simplify contentElement.editable ViewHelper

Old template code:

```
1<neos:contentElement.editable property="text">{text -> f:format.raw()}</neos:contentElement.editable>
```

New template code:

```
1{neos:contentElement.editable(property: 'text')}
```

(For B/C reasons the contents of this tag still override the property, if present)

2) Avoid wrapping tag of contentElement.editable ViewHelper in Frontend

Given the property text contains the string

```
1<p>Lorem ipsum</p>
```

Old output (Backend):

```
1<div property="typo3:text" class="neos-inline-editable">
2 <p>Lorem ipsum</p>
3</div>
```

New output (Backend):

```
1<neos-editable-property data-neos-property="text">
2 <p>Lorem ipsum</p>
3</neos-editable-property>
```

(We still need to test, if custom tags work with Aloha and all supported browsers. Otherwise this should probably stay a div, possibly with some CSS that removes visual impact (display: inline, margin: 0, padding: 0, ...))

Old output (Frontend):

```
1<div>
2 <p>Lorem ipsum</p>
3</div>
```

New output (Frontend):

```
1<p>Lorem ipsum</p>
```

(Note that there is "no wrapping tag" added by the ViewHelper any longer)

3) Deprecate contentElement ViewHelper

The attributes that are required for the backend to work (data-neos-*...) are added by a TypoScript processor (only in backend):

Old template code:

```
1<neos:contentElement node="{node}">
```



```
2 ...
3</neos:contentElement>
```

New template code:

```
1...
```

(For B/C reasons the ViewHelper still renders a tag - but it has no advantage over hard-coding that tag in the template itself)

TODO:

- Introduce new **Content** TS Object: prototype(TYPO3.Neos:Content) < TYPO3.TypoScript:Template
- Deprecated **TYPO3.Neos:Template**: prototype(TYPO3.Neos:Template) < TYPO3.Neos:Content // DEPRECATED
- Make Plugin(View) extend **Content**: prototype(TYPO3.Neos:Plugin) < TYPO3.Neos:Content
- Add "meta data" processor to **Content**

Processor Chain:

- add outermost wrapping div IF IT DOES NOT EXIST.
 - configuration option for disabling this in live workspace. by default same markup in live + user WS.
- add Node Type Name -> CSS Class Name generator
 - no config options. Neos should not rely on class names, but data attributes.
- add "ID" generator
 - configuration option for disabling this. by default same markup in live + user WS.
- "Backend" data-attribute generator
 - re-use outermost wrapping element if it exists
 - have class name NOT hardcoded in template

#6 - 2013-12-04 18:41 - Bastian Waidelich

- Status changed from *Accepted* to *Resolved*
- Target version set to *1.0 beta 2*
- Estimated time set to *0.00*

#7 - 2013-12-04 18:43 - Bastian Waidelich

- Estimated time deleted (*0.00*)