

Redmine - Patch #35104

Code blocks - consistent rendering and retaining user-supplied language name in rendered HTML

2021-04-15 21:29 - Martin Cizek

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Marius BĂLTEANU	% Done:	0%
Category:	Text formatting	Estimated time:	0.00 hour
Target version:	5.0.0		

Description

This patch:

- Makes code block rendering consistent across formats, which is important for plugins, client-side scripts and styling.
- Makes sure rendered language name is escaped.
- Exposes user-supplied language name in an HTML attribute, so that client-side scripts and Redmine plugins can always access it.

The patch applies to the current Textile and Markdown formatters. The very same change to the planned CommonMark formatter will be attached to [#32424](#).

Please see below for design decisions rationale.

Code block rendering

<pre><code> vs. just <pre>:

Textile supports blocks without code (<pre>...</pre>), while code can be part of such blocks. Typically, there is just one and only code block per <pre> block, but it's possible to have more than one code block in a <pre> block. So in general, we cannot avoid the code tag in the HTML rendered from Textile.

While it is possible to render only the pre tag for Markdown code blocks and rendering libraries do have option for that, we should use the code tag too to be consistent. Fortunately, this is already the case and it is even what CommonMark spec suggests.

No-code blocks in Markdown

Markdown does not have preformatted blocks except for code blocks. Code blocks are either indented or fenced, but both are meant to represent code ([spec](#)).

The fenced code blocks can also carry an info string, which is used to determine the language of the code.

Comparison of different renderers:

Formatter	No-code block → HTML	Code block w/o lang → HTML	Code block with supported lang → HTML	Code block with unsupported lang → HTML
Redmine Textile	<pre>...</pre>	<pre><code>...</code></pre>	<pre><code class="ruby syntaxhl">...</code></pre>	<pre><code>...</code></pre>
CommonMark reference renderer	N/A	<pre><code>...</code></pre>	<pre><code class="language-ruby">...</code></pre>	<pre><code class="language-foo">...</code></pre>
Redmine Markdown	N/A	<pre>...</pre>	<pre><code class="ruby syntaxhl">...</code></pre>	<pre>...</pre>
Redmine CommonMark #32424	N/A	<pre><code>...</code></pre>	<pre><code class="ruby syntaxhl">...</code></pre>	<pre>...</pre>

It's quite obvious that we should follow the Textile behavior to be consistent.

Original data language

The user-supplied language information is currently dropped if the particular language does not have any lexer for syntax highlighting. However, there are several use cases when this information is valuable:

- Plugins like [redmine_wysiwyg_editor](#)
- Data format converters like [redmine_reformat](#)
- Client-side scripts that could render or decorate specific languages in their own way, e.g. [mermaid](#)

For the reasons above, the information has to be carried on the code tag. I've examined a few projects and libraries related to the topic to find an appropriate name for such attribute. The conclusion is to use `<code data-language="foo" ...>`.

Resulting code block rendering:	Formatter	No-code block → HTML	Code block w/o lang → HTML	Code block with supported lang → HTML	Code block with unsupported lang → HTML
	All Redmine formatters	<code><pre>...</pre></code>	<code><pre><code>...</code></pre></code>	<code><pre><code class="ruby syntaxhl" data-language="ruby">...</code></pre></code>	<code><pre><code data-language="foo">...</code></pre></code>

Quality Assurance, Security & Compatibility

- **QA:** Added new unit tests and updated the existing ones.
- **Security:** Previously, Textile formatter did not escape language name and relied on validating it against syntax highlighter, while expecting language names to be safe. Language names are now always subject to escaping.
- **Compatibility:**
 - The rendered HTML is now the same as the HTML from Textile.
 - The added data-language attribute should be indeed ignored by unconcerned components.
 - I don't really expect that anybody to rely on getting `<pre>` and not `<pre><code>`, especially when it is the current behavior of Textile. I've verified this on `redmine_wysiwyg_editor` as a rare example of a potentially concerned plugin.

So I'm convinced it is safe to include this in any minor release.

Related issues:

Related to Redmine - Feature #32424: CommonMark Markdown Text Formatting

Closed

Associated revisions

Revision 21182 - 2021-08-15 22:59 - Marius BĂLTEANU

Add "data-language" attribute to code block with the user-supplied language for CommonMark formatter (#35104, #32424).

Patch by Martin Cizek.

Revision 21183 - 2021-08-15 23:00 - Marius BĂLTEANU

Unify code block and add "data-language" attribute with the user-supplied language for Textile and Markdown formatters (#35104).

Patch by Martin Cizek.

History

#1 - 2021-04-15 22:00 - Martin Cizek

- File 0002-application_helper_test.patch added

#2 - 2021-04-15 22:02 - Martin Cizek

...accidentally marked this as a defect, it should have been patch. Thank you.

#3 - 2021-06-03 14:46 - Go MAEDA

- Tracker changed from Defect to Patch

#4 - 2021-08-15 13:10 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

- Target version set to 5.0.0

#5 - 2021-08-15 14:36 - Marius BĂLTEANU

- Related to Feature #32424: CommonMark Markdown Text Formatting added

#6 - 2021-08-16 00:24 - Marius BĂLTEANU

- Status changed from New to Closed

Patches committed, thanks again Martin for your work on this.

Files

0001-unify-code-blocks.patch	6.37 KB	2021-04-15	Martin Cizek
0002-application_helper_test.patch	1.78 KB	2021-04-15	Martin Cizek