

## Redmine - Feature #14936

### Use a stable and modifiable data structure for jsToolBar elements

2013-09-18 15:40 - Felix Schäfer

<b>Status:</b> New	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>Resolution:</b>	
<b>Description</b>	
<p>The elements of the jsToolBar are currently stored in a javascript {} (see <a href="source:/trunk/public/javascripts/jstoolbar/jstoolbar.js@12152#L153">source:/trunk/public/javascripts/jstoolbar/jstoolbar.js@12152#L153</a>) and are read in the hopes they will come out in the same order they have been put in (see <a href="source:/trunk/public/javascripts/jstoolbar/jstoolbar.js@12152#L213">source:/trunk/public/javascripts/jstoolbar/jstoolbar.js@12152#L213</a>). This however is not a specified behavior of javascript, it just happens to work in most browsers (<a href="#">Chrome being the exception here in certain cases</a> (sorry for the weird anchor of the link, the enclosing section has none)). The specification explicitly states:</p> <p>The [...] order of enumerating the properties [...] is not specified.</p> <p>(see <a href="#">ECMA-262 12.6.4</a>)</p> <p>We (<a href="#">Planio</a>) haven't seen any bugs with the current implementation but find it lacking in that we cannot add a button at a specific position in the toolbar without adding it to core js files (which has become more difficult since the jstoolbar-textile.min.js is shipped with core without any apparent clues how it is generated, but that's not the point here).</p> <p>Would the Redmine team be open to switching the aforementioned jsToolBar.prototype.elements to a data structure with a guaranteed ordering and the ability to add elements at a specified absolute ("at position X") or relative ("after button X") place?</p>	
<b>Related issues:</b>	
Related to Redmine - Feature #14937: Code highlighting toolbar button	<b>Closed</b>

#### History

##### #1 - 2013-09-18 15:51 - Felix Schäfer

(Note: this is related to [#14937](#) but the button submitted in [#14937](#) is not the only use-case we have for this, so fixing [#14937](#) doesn't fix this)

##### #2 - 2013-09-29 09:52 - Toshi MARUYAMA

- Related to Feature #14937: Code highlighting toolbar button added

##### #3 - 2013-10-27 04:47 - Mischa The Evil

Felix Schäfer wrote:

[...] but find it lacking in that we cannot add a button at a specific position in the toolbar without adding it to core js files [...]

I agree on that.

[...] switching the aforementioned jsToolBar.prototype.elements to a data structure with a guaranteed ordering and the ability to add elements at a specified absolute ("at position X") or relative ("after button X") place?

To me this sounds like a good idea. This will give plugin developers (thinking about custom macros, custom syntax, etc.; like the often used WikiNG plugin) the control they need.

##### #4 - 2013-10-27 14:38 - Yukinari TOYOTA

I have an experience of similar problem.

In that case, I want to add a special image button after the original image button.

I solved this problem by modifying/replacing `jsToolBar.prototype.elements` at runtime.

However, it is not straightforward so new API is welcome.

see: [https://github.com/t-yuki/clipboard\\_image\\_paste/commit/122a9134165ecc3c6a056fdab788efd2d95bfb3](https://github.com/t-yuki/clipboard_image_paste/commit/122a9134165ecc3c6a056fdab788efd2d95bfb3)