

Capturing content from nicEdit when form is submitted by Javascript

nicEdit is a very lightweight wysiwyg HTML editor, distributed under the MIT license (which means that as long as the copyright info remains intact, you can do pretty much anything with it, including distribute it with a commercial application). There are a couple of drawbacks to nicEdit - first, it does not produce standards compliant HTML. If it did, it would not be anywhere near as lightweight, so that's a bit of a tradeoff (it relies on the browser's own HTML editing features, so if you use a decent browser you should get pretty good results). The other main problem I found though, is that if you have a form containing a nicEdit editor, and submit the form using Javascript ie. `form.submit()`, the content of the editor is not posted back to the server!

If you would like to know why that is, and what you can do about it, read on...

The root cause of the problem is that the Javascript `onsubmit` event for form elements is only triggered by a user clicking on a submit button. Most browsers do not trigger the `onsubmit` function when a form is submitted by Javascript. This is insane, counter-intuitive, and a pain in the neck, but we have to live with it.

The normal solution to the problem then, is to call whatever code the `onsubmit` event calls before submitting the form. That's all very well if you are writing the code yourself, but if you are using nicEdit, and your knowledge of Javascript is a bit patchy, you could find yourself a bit stuck. Even looking at the generated source of your form will not show you what is happening in the form's `onsubmit` event because nicEdit programatically adds an event listener to the form. So I had to delve into the nicEdit source code to find out what to call.

Assuming you have instantiated a nicEdit editor and attached it to a textarea on your form using a piece of code similar to this:

```
var editor1 = new nicEditor({iconsPath : '/path/to/nicEdit/nicEditorIcons.gif', fullPanel : true}).panelInstance('text_area_id');
```

...you would then need to do this to capture the HTML into your textarea before submitting the form via javascript:

```
for(var i=0;i<editor1.nicInstances.length;i++){editor1.nicInstances[i].saveContent();}
```

I used a for loop because `nicInstances` is an array. As far as I can tell, there will only ever be one element in that array when nicEdit is instantiated as shown above, but just to be sure I iterate through all elements. After calling that `saveContent` method, the HTML should be copied into the textarea and therefore available in the postback. Of course if you have more than one editor on a page, you would have to repeat it for each editor.

