

Stopping form submissions containing HTML code

Posted At : 15 July 2013 06:10 | Posted By : Shaun McCran
Related Categories: JQuery, Javascript

Are you getting web form spam? I was, mainly from bots, or 'copy and paste' farms where users sit and paste bulk code into your forms. Mostly these contain links or image tags. Rather than compromise the form usability for genuine users I decided to fix my spam issue with a little more tact than you may usually find, such as in the case of Captcha devices.

I'm already cleaning html code submissions on the server side but why not make things even more informative for my users and tell them up front that their HTML isn't appreciated.

I didn't want a complicated Regex or pattern matching plugin, I simply wanted to detect key HTML elements and tell the user that it wasn't going to be accepted by the form. This code uses the JQuery plugin for form validation. You can get it here:

<http://jqueryvalidation.org/>

First things first, let's create a custom validation rule to detect our html elements.

```
</script language="javascript">
$(document).ready(function(){

$.validator.addMethod("findhtml", function(value, element) {

    var commentclean = true,
        disallowed = ['<', 'http'];

    for (var i=0, len = disallowed.length; i<len; i++) {
        if (element.value.indexOf(disallowed[i]) != -1) {
            commentclean = false;
            break;
        }
    }

    return this.optional(element) || ( commentclean );
});
});
```

```
    }, "* match the string");  
    });  
</s/crript>
```

This creates an array of disallowed elements and loops through them when the rule is invoked.

Secondly we will use this rule in our validation routine when a user tries to submit the form.

```
<!-- Form now powered from JQuery --->  
<s/crript type="text/javascript">  
    $(document).ready(function(){  
        $("#form").validate({  
            rules: {  
                name: {required: true},  
                email: {required: true},  
                tel: {required: false},  
                message: {required: true, findhtml: true}  
            },  
            messages: {name: "Please enter your Name",  
                email: "Please enter a valid email address",  
                tel: "",  
                message: {required: "Please enter a message", findhtml: "You cannot ente  
            }  
        });  
    });  
</s/crript>
```

The key line here is:

```
message: {required: true, findhtml: true}
```

This invokes our previously created validation rule.

In this way the user is told 'You cannot enter HTML in this field'. A friendly validation message that clearly shows WHY the form isn't going to work.

You can see this working on my contact form here : <http://www.mccran.co.uk/contact/>