

Validating checkboxes using the JQuery validate plug-in

Posted At : 16 March 2010 11:49 | Posted By : Shaun McCran
Related Categories: JQuery, Javascript, RIA

I have recently spent some time building a dynamic XML driven form generator. The last element of this was the checkbox form element. All my other fields were being validated as either required, or needing a certain length using the JQuery Validation plug-in. (<http://jquery.bassistance.de/validate/>)

This article explains how I put together a checkbox validation routine using the Validate plug-in.

Start of by including the JQuery library and the Validation plug-in.

```
<html>
<head>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
<script type="text/javascript" src="jquery.validate.pack.js"></script>
```

Next create a small form with your checkbox values in. In the example below I am listing a series of pie flavours (I like Pie). They all have the same 'name' attribute. There is also a final option for users who wish to enter their own values.

```
<style type="text/css">
body{padding:0; margin:10; font-family:Arial, Helvetica, sans-serif; font-size:12px}
#flavour label.error {display: none;}
label.error {color:red; padding: 5px;}
</style>
</head>
```

```

<body>

  <h1>What is your favourite type of Pie?</h1>

  <form method="post" id="form">
    <fieldset id="flavour">
<label for="flavour" class="error">Please select at least one flavour.<br></label>
      <input name="flavour" id="1" type="checkbox" value="1"/>Ham & Mushroom<br/>
      <input name="flavour" id="2" type="checkbox" value="2"/>Steak & Ale<br/>
      <input name="flavour" id="3" type="checkbox" value="3"/>Chicken<br/>
      <input name="flavour" id="4" type="checkbox" value="4"/>Vegetarian<br/>
      <input name="flavour" id="other" type="checkbox" value="5"/>Other<br/>
      <p></p>
      If Other<br><input type="text" name="otherFlavour">
    </fieldset>
  <p><input class="submit" type="submit" value="Submit"/></p>
  </form>
</body>
</html>

```

Lastly we must construct the Validate plug-in script. The script attaches itself to the id 'form', and validates against the 'flavour' fields. By declaring that 'flavour: { required : true}' we are saying that the form element 'flavour' must exist. Also the 'minlength' declaration here can be used to determine how many of the checkbox values are required. Changing it to 3 will force the user to make 3 selections.

The 'otherFlavour' requirement is a little trickier. It is required based on the condition of the '#other' field(id) having been selected. Only then will the validation rules for that element be run. In this case a length check to make sure it has a value.

```

</script type="text/javascript">
  $.ready(function() {
    $("#form").validate({
      rules: {
        flavour: {required: true, minlength: 1},
        otherFlavour: { required: function(element){
          if ($('#other').is(':checked')) {
            return true;
          } else {
            return false;
          }
        },
          minlength: 1
      }
    }
  }

```

```
        },  
        messages: {otherFlavour: "Please specify a different flavour."  
    });  
});  
</script>
```

Using this code it is possible to enforce client side checkbox validation.
You can see an example of this [here](#) .