

## Using JQuery to dynamically add form fields to a page

Posted At : 18 August 2011 14:51 | Posted By : Shaun McCran  
Related Categories: JQuery, Javascript

I'd seen a few different ways of dynamically adding form fields to a web page but hadn't ever tried it. I was interested in seeing how exactly you can do this at a technical level, but also how you can manage the users interactions, allowing them to keep adding fields as they complete the previously added ones.

Firstly lets define a few goals:

1. I want to add new form fields to an existing form, and they must have unique ids / names.
2. I want to make sure that a user can keep adding more new fields but only after they have completed the previous added field.

Let's setup the form first.

```
<div class="form-label">
  <label for="static">Static form field</label>
</div>
<div class="form-field">
  <input type="text" name="static" id="static" size="20">
</div>

<div id="newfields"></div>

<div id="more-link" class="clear-both big"></div>

<div class="form-label"></div>
<div class="form-field">
  <input type="submit" name="Action" value="submit">
</div>
```

The first form field is a 'normal' static field, nothing exciting going on here.  
Next there is a div element "newfields" this is a container for our newly added fields.

Next we have a div "more-link" this contains a JQuery powered link that users will click on to add new fields. I'm creating this in JavaScript so that I can turn the link on and off based on some conditional logic.

Now lets take a look at the JQuery that will populate these divs.

```

</script>
// create an index value
var index = 1;

// adds a new field
function addfield() {
    index++;
    appendStr = '<div class="form-label"><label for="newfield'+index+'">New field no '+:
    '<div class="form-field"><input id="newfield'+index+'" name="newfield'+index+'" s

    $('#newfields').append(appendStr);
    // turn off the click event to create new fields
    removeStr();
}

// disables the hyperlink
function removeStr(){
    $('#more-link').html('Add field')
}

// enables the hyperlink
function addStr(){
    $('#more-link').html('<a href=javascript:void(0); id=add-field>Add field

$(document).ready(function(){

    // create the link when the dom loads
    addStr();

    // use live() to create a click event
    $('#add-field').live('click', function() {
        addfield();
    });

    // did the user click out of the new field, does it have a value?
    $('#.exit-detect').live('blur', function() {

        // check if the field length is gt 0

```

```
        if($(this).val().length > 0){
            // turn the link back on
            addStr();
        }

    });

    });
</script>
```

There is a fair bit of code here, so we'll break it down. Initially I create an index value. This will be incremented each time we create a new field.

Next we have a function 'addfield', this creates the form elements and appends them into the div 'newfields'. It then disables the hyperlink that creates the fields.

Next there are two functions 'removeStr' and 'addStr' these create or destroy a hyperlink to allow users to make new fields. They are called from other functions here.

Now we are inside our document.ready, so this is going to fire when the DOM has loaded. Firstly we create the link to add fields using 'addStr()'.

Next there is a click event on the selector '#add-field'. This runs the function (addfield) that creates the form fields.

Lastly there is a selector on the element 'exit-detect' that runs a blur event on the form field that was created. It checks that the newly created field has a value in it, if it does then it re-enables the hyperlink, allowing the user to create more fields. In this way a user cannot just click 'add' a dozen times and create a load of blank fields.

Note that most of these selectors are using the live() method to bind an event as the dom elements don't exist on page load.

[Follow this link to view a demo of using JQuery to dynamically add form fields](#) .