Server-side Vs. Client-side Validation Revisited

Posted At : March 16, 2006 10:11 AM | Posted By : Mark Kruger
Related Categories: Coldfusion MX 7, Coldfusion Tips and Techniques

> CF Muse Reader Asks:
> You suggest both client and server side should be used for validation. Just to check - I should code so that js picks up the errors first using event code or onsubmit then let server side pick up the errors on submit using cfinput validate/required etc. Or do I recode all the stuff to be event driven such as onchange etc.

This question refers to the previous post on form validation. From the way the question is phrased I believe the reader is probably proficient at JavaScript. There are some amazing things you can do with JavaScript, and I'm in favor of creating intuitive interfaces that help your user cope with the complexity of your application. Obviously JavaScript is or can be a big part of that solution. If your concern is to provide a better user experience then JavaScript is helpful and necessary. If your concern is to *validate your data for accuracy and security* then you *must* use server side code to check your form inputs.

As an example, you might have a form like this:

```
<form action="myForm.cfm" method="post" onsubmit="validate()">
   <input name="myName" type="text">
   <input name="myDateOfBirth" type="text">
   </form>
```

Let's say JS function "validate()" checked to make sure "myDateOfBirth" was actually a date and was in the last 120 years before submitting. That would be useful for the user if he happened to type in 1/12/198 instead of 1/12/1982. If, however, JS is turned off he may be able to submit the form anyway - regardless of the input. More importantly, if this is an important form and the user is up to no good, he can simply re-create your form on his own web site and point it to your domain - firing post requests willy nilly till he sees a hole. In fact, let's say the form above was in a folder called "forms" on "mydomain.com". This snippet of code would allow him to circumvent your clever JavaScript with no problem.

```
<cfhttp url="http://mydomain.com/forms/myform.cfm" method="post">
      <cfhttpparam type="FORMFIELD" name="myName" value="Attila The Hun"/>
      <cfhttpparam type="FORMFIELD" name="myDateOfBirth" value="02/05/406"/>
   </cfhttp>
```

Server Side Comes First

The point is, that *regardless of what you do on the client* you should validate the inputs on the server. On a public form you have little or no control over what is happening on the client. All you can *really* control is what goes into you application or database.

Now, the second part of the question asks about "cfinput" and "validate/required". To be clear, when you use CFFORM you are not creating server side code. You are, in fact creating complex JavaScript routines for validation that function on the client side - so you are *not using* server side validation. On a side note, I really detest cfform and tend to avoid it in favor or custom JS or something like Dan Switzer's "Qforms". Unless I'm using flash forms I avoid cfform altogether. One additional approach provided by

Coldfusion that some folks use is the "hidden variable" approach. The muse reader above may be referring to this idea. It's actually a rudimentary method for server side validation. Here's how it works. You would alter your form as follows:

```
<form action="myForm.cfm" method="post" onsubmit="validate()">
   <input name="myName" type="text">
   <input name="myName_required" type="hidden" value="You must provide a name">
   <input name="myDateOfBirth" type="text">
   <input name="myDateOfBirth_date" type="hidden" value="You must provide a valid date">
   </form>
```

Cleverly, Coldfusion looks for the "_required" or "_date" (or _integer or _float etc. - see the docs) that matches the name of a form element and then validates that element on the server - displaying the message you have chosen to the user and asking them to go back and fix the problem. Yes, this is server side validation because the *server is doing the checking*, but is it secure from malicious attack? Nope. All a nere-do-well has to do is remove the "myDateOfBirth_date" form element from his custom form or CFHTTP call. The server's validation is *based on these inputs being present*. If those form elements are not part of the request, the server will not check them.

What is Server Side

Server side validation is the process of painstakingly checking the inputs and making sure they contain what they are supposed to contain. Most of my server side validation code looks like this:

```
<cfscript>

// default values
err   =   0;
   msg   =   '';

// check for name (isEmpty checks "trim(len(arg))"
if(isEmpty(form.myName)) {
      err      =   1;
      msg      =   '<p>You must provide a name</p>';
   }

// check for date
if(isEmpty(form.myDateOfBirth)
      OR NOT isDate(form.myDateOfBirth)
         OR dateDiff('yyyy',form.myDateOfBirth,now()) GT 120) {
      err      =   1;
      msg      =   '<p>You must provide your date of birth
                  and you must be living</p>';
   }

// if not err pass the args
if(NOT err)
      msg      =   resourceCfc.addUser(argumentcollection=duplicate(form));

</cfscript>
```

As you can see I've taken the time to specify individual form elements and conditions in my handler.

In my opinion, if security and data integrity are your primary concerns, you should *start* with server side validation and put it above client side validation in importance. You should add client side validation to reduce round trips to the server and maximize the user experience but don't rely on it to keep your site safe or your data sound.