

Getting Started with VT - Enabling Client-Side Validations

Posted At : April 29, 2009 8:45 PM | Posted By : Bob Silverberg

Related Categories: ColdFusion, Getting Started with VT, ValidateThis

In the [previous post](#) in this series about getting started with ValidateThis!, my validation framework for ColdFusion objects, we looked at how to very quickly get up and running with the framework, and how to use it to perform server-side validations for our objects. In this post we'll look at enabling client-side validations for those same business rules.

VT is designed to generate both server-side and client-side validations from a single set of business rules, so we don't have to do any additional coding for the validations themselves, all we have to do it a bit of set up, and then ask the framework for the JavaScript. Let's start with the set up.

Step 1 - The JS Files

By design, VT can support any JavaScript validation library, including one of your own making, but it currently ships with an implementation for a single validation library, the [jQuery Validation Plugin](#). VT will generate JavaScript statements for all of your validations, and those statements will assume that a few JavaScript files have been loaded on your page. The following files are required to allow the jQuery Validation Plugin to work:

- jquery-1.3.2.min.js - the jQuery library
- jquery.validate.pack.js - the jQuery Validation Plugin
- jquery.field.min.js - the [jQuery Field Plugin](#)

The latest versions of those files ship with VT and can be found in /ValidateThis/client/jquery/js. So, you'll need to copy those files to a web accessible folder (e.g., /js/), if you don't already have a copy of them from whence they can be accessed.

Step 2 - Load JavaScript Libraries

As discussed above, those files need to be loaded into your web page. You can do that manually, using code similar to this:

```
<script src="/js/jquery-1.3.2.min.js" type="text/javascript"></script>

<script src="/js/jquery.field.min.js" type="text/javascript"></script>

<script src="/js/jquery.validate.pack.js" type="text/javascript"></script>
```

Or you can ask VT to load the files for you, using a convenience method. Assuming you have access to VT in a variable, such as application.ValidateThis (as discussed in the [previous post](#)), you can simply call getInitializationScript(), passing in the objectType:

```
<cfhtmlhead text=

    "#application.ValidateThis.getInitializationScript(objectType='user.user')#"

/>
```

VT will generate the code required to load all of the libraries for you, which cfhtmlhead then places into your head tag.

Step 3 - Ask VT to Generate the Validation JavaScript

Similar to the above, you can simply call getValidationScript(), passing in the objectType:

```
<cfhtmlhead text=

    "#application.ValidateThis.getValidationScript(objectType='user.user')#"

/>
```

VT will generate individual JavaScript statements to implement all of your object's validations.

And it's as simple as that - your page will now provide client-side validations!

In the next post I'll look at some of the other validation types that are currently available in the framework.

As always, if you have any questions or interest in following the progress of VT, I invite you to join the [ValidateThis! Google Group](#).