

Contents

- [Generating Client-Side Validations](#)
 - [The `getInitializationScript\(\)` Method](#)
 - [Arguments](#)
 - [Return Value](#)
 - [Examples](#)
 - [The `getValidationScript\(\)` Method](#)
 - [Arguments](#)
 - [Return Value](#)
 - [Example](#)

Generating Client-Side Validations

There are two methods that you'll use when generating client-side validations:

The `getInitializationScript()` Method

Use the `getInitializationScript()` method to return JavaScript code to set up client-side validations. You only need to generate this script once per page, and can include links for the JavaScript libraries required by the framework.

Arguments

It accepts the following arguments:

- *JSLib* (optional) The name of the JavaScript implementation to be used to generate the script.
 - If not passed in the *JSLib* that is specified in the *DefaultJSLib* key of the [ValidateThisConfig Struct](#) will be used.
 - Note that as there is currently only one JS implementation available, it is currently unnecessary to pass this argument into this method.
- *JSLIncludes* (optional) A boolean indicating whether to return the JS statements that include the libraries required by the framework.
 - This exists to allow a developer to generate the setup script, but still manually add their own `<script>` tags for the libraries. This can be helpful when one is already including some of the libraries (e.g., jQuery) in their document.
- *locale* (optional) The locale to be used for translating validation failure messages.
 - Unless you are supporting multiple languages this argument can be ignored.

Return Value

The `getInitializationScript()` method returns a string that contains an entire block of JavaScript which can then be inserted into your page. This includes the opening and closing `<script>` tags.

Examples

```
<cfset theScript = application.ValidateThis.getInitializationScript() />
<cfhtmlhead text="#theScript#" />
```

This would ask the framework to generate the JavaScript required to setup the client-side validations. That script is then being loaded into the document via the `<cfhtmlhead>` tag.

If one does not want the framework to include the `<script>` tag to load the required libraries, one would call it like this:

```
<cfset theScript = application.ValidateThis.getInitializationScript(JSLIncludes= false) />
```

The `getValidationScript()` Method

Use the `getValidationScript()` method to return JavaScript code for all of the rules that you've defined.

Arguments

It accepts the following arguments:

- *theObject* (optional) The actual object for which to generate client-side validations.
- *objectType* (optional) The name of the object type as defined to the framework.
 - One of either *theObject* or *objectType* must be specified, so the framework can identify for which object the script should be returned.
 - For more information on why both are optional but at least one is required, see the section on [Specifying the objectType on a Method Call](#).
- *context* (optional) The name of a context used in the object's rules definition file.
 - If passed, the object will be validated using the specified context. Only rules assigned to that context will be evaluated.
- *formName* (optional) The name of the form in the html document to which validations are to be attached.
 - If not passed in the framework will use one of two values:
 - If a *formName* is associated with the given *context* in your [Rules Definition File](#) via the `<contexts>` element, that *formName* will be used.
 - Otherwise the *formName* that is specified in the *defaultFormName* key of the [ValidateThisConfig Struct](#) will be used.
- *JSLib* (optional) The name of the JavaScript implementation to be used to generate the script.
 - If not passed in the *JSLib* that is specified in the *DefaultJSLib* key of the [ValidateThisConfig Struct](#) will be used.
 - Note that as there is currently only one JS implementation available, it is currently unnecessary to pass this argument into this method.
- *locale* (optional) The locale to be used for translating validation failure messages.
 - Unless you are supporting multiple languages this argument can be ignored.

Return Value

The `getValidationScript()` method returns a string that contains an entire block of JavaScript which can then be inserted into your page. This includes the opening and closing `<script>` tags.

Example

```
<cfset theScript = application.ValidateThis.getValidationScript(objectType= 'User', context= 'Register' ) />
<cfhtmlhead text="#theScript#" />
```

This would ask the framework to generate the JavaScript required to perform the client-side validations defined in the rules definition file called *User.xml* for the context of *Register*. That script is then being loaded into the document via the `<cfhtmlhead>` tag.

For more information on interacting with the `ValidateThis.cfc` facade object, see the section on [Using the ValidateThis Facade Object](#).