

Contents

- [cfAjaxProxy](#)
- [Syntax](#)
- [Description](#)
- [Body](#)
- [Attributes](#)
- [Javascript Api](#)
- [Example Usage](#)
- [Create a proxy object](#)
- [Use the proxy object instance](#)
- [Binding Syntax](#)
- [Bind Javascript](#)
- [Bind Cfc](#)
- [Bind Url](#)

Return to [Tags](#) or [Functions](#)

cfAjaxProxy

Syntax

```
[bind="string"]  
[cfc="string"]  
[extends="boolean"]  
[jsclassname="string"]  
[methods="string"]  
[onerror="string"]  
[onsuccess="string"]>  
[]
```

[] = Optional attribute

Description

Creates a JavaScript proxy for a component, for use in an AJAX client. Alternatively, creates a proxy for a single CFC method, JavaScript function, or URL that is bound to one or more control attribute values.

Body

This tag may have a body.

Attributes

The attributes for this tag are fixed. Except for the following attributes no other attributes are allowed.

Name	Type	Required	Description
bind	string	No	A bind expression that specifies a CFC method, JavaScript function, or URL to call. the CFC for which to create a proxy. You must specify a dot-delimited path to the CFC. The path can be absolute or relative to location of the CFML page. For example, if the myCFC CFC is in the cfc subdirectory of the Railox page, specify
cfc	string	No	cfc.myCFC. On UNIX based systems, the tag searches first for a file whos name or path corresponds to the specified name or path, but is in all lower case. If it does not find it, Railox then searches for a file name or path that coresponds to the attribute value exactly, with identical character casing.
extends	boolean	No	
jsclassname	string	No	The name to use for the JavaScript proxy class.
methods	string	No	
onerror	string	No	The name of a JavaScript function to invoke when a bind, specified by the bind attribute fails. The function must take two arguments: an error code and an error message. The name of a JavaScript function to invoke when a bind, specified by the bind attribute succeeds. The function must take one argument, the bind function return value. If the bind function is a CFC function, the return value is automatically converted to a JavaScript variable before being passed to the
onsuccess	string	No	onSuccess function.

Javascript Api

The followign method can be called on a javascript proxy object instance.

Method	Description
setAsyncMode()	As per default the remote proxy object calls will run in asynch mode.
setCallbackHandler(function)	Function called as callback. The function takes 2 arguments: the value returned and the textStatus of the ajax call.
setErrorHandler(function)	Function called as error callback. The function takes 2 arguments: the error code and the error message.
setHTTPMethod("method")	GET or POST (default is GET) Says to the proxy obejct how to parse the result of the call to the cfc method:
setReturnFormat(format)	<ul style="list-style-type: none"> • json (default) • plain (plain text) • xml - result is an xml that will be parsed into a js object.
setSyncMode()	Convert call in synch mode. Browser will wait the ajax response to proceed.
setForm('id')	Serialize the form with the provided id and append it to the query string. Specifies the JSON format in which to return ColdFusion query data. The parameter must have one of the following values:
setQueryFormat('format')	<ul style="list-style-type: none"> • row (default) - Sends the data as a JSON object with two entries: the column names and an array of row arrays. • column - Sends the data as a JSON object that represents WDDX query format. This object has three entries: the number of rows, an array with the column names, and an object where the keys are the column names and the values are arrays containing the column data.

Example Usage

Create a proxy object

```
cfc="ajaxproxy.cfc.test" jsclassname="proxyObj" onSuccess="successCallback" onError="errorCallback"/>
```

Use the proxy object instance

```
<script></span> type="text/javascript">
var myProxy = new proxyObj();
myProxy.getData();
/* call a amethod passing parameters */
var myProxy = new proxyObj();
myProxy.getData(200, 'text');
/* You can also pass parameters like a js literal object*/
var myProxy = new proxyObj();
var args = {arg:100,arg2:200};
myProxy.getData(args);
</script>
```

Binding Syntax

Visit [this page](#) for more about the supported binding syntax.

Bind Javascript

```
bind="javascript:updateDiv({myForm:myName},{myForm:myAge})" />
```

When element with name "myAge" and "myName" included into a form with id "myForm" change the js function "updateDiv" is invoked. The function receive as argument a js literal object that contains the actual values of the elements binded.

Bind Cfc

```
bind="cfc:mysite.mycfc.method({myForm:myName},{myForm:myAge})"  
onSuccess="onSuccess" onError="onError"/>
```

Bind Url

```
bind="url:file.cfm?name={myForm:myName}&age={myForm:myAge}"  
onSuccess="onSuccess" onError="onError"/>
```