

Model-Glue 2 Cheat Sheet

Event API

When a function in a controller is called as a result of a <message-listener> tag, it is passed an instance of the ModelGlue.unity.eventrequest.EventContext CFC. This CFC contains a series of functions known as the Event API that are useful to getting and setting values in the viewstate as well as event-related utilities.

Function	Arguments	Returns
AddResult(name:string) Adds a result by the given name. If a <result> tag exists in the currently executing <event-handler> whose NAME attribute matches the value passed, the <event-handler> declared in its DO attribute will be added to the request.	Name (Required) - The name of the result to add	Void
ArgumentExists(string:name) Does a value of the given name exist in the current message tag's arguments block?	Name (required) - The name of the argument to check	Boolean
Forward(eventhandler:string [, append:list]) Immediately redirects the request to the event handler named. All values in the viewstate are maintained across the redirect if session is enabled.	EventHandler (Required) - The name of the event handler to redirect to Append (Optional) - Will append the list of values passed in append to the resultant URL.	Void.
GetAllArguments() Returns the structure that contains all argument members by reference.	None	Struct
GetAllValues() Returns the structure that contains all viewstate members by reference.	None	Struct
GetArgument(name:string, [default:any]) Returns a value from the current message tag's arguments block.	Name (required) - The name of the argument to retrieve Default (optional) - If the argument does not exist, a default value to return. It is not set into the argument collection - the collection is read-only.	Any
GetValue(name:string, [default:any]) Returns a value from the viewstate, such as a form or url variable.	Name (required) - The name of the value to retrieve Default (optional) - If the value does not exist, a default value to set into the viewstate and then return	Any
GetView(name:string) Returns the rendered content of a view.	Name (required) - The name of the view to retrieve	String
MakeEventBean(type:any, [fields:list]) Loops over a CFC instance, looking for any methods whose names start with "set." For any "setter" methods where a like-named value exists in the viewstate, the setter will be called, passing this value.	Type (Required) - If a string is passed, it will instantiate a CFC of the type passed and attempt to call a function named "init" if one exists. If a CFC instance is passed, it will use the instance. Fields (Optional) - If a list of fields is passed, it will only attempt to set the properties listed.	CFC Instance
Trace(name:string, value:any) Adds the value to the debugging trace and gives it the name passed. If a complex value is passed, the result of its <cfdump> is displayed.	Name (Required) - A label to identify the value in the debugging trace Value (Required) - The value to display in the debugging trace. If a complex value is passed, the result of its <cfdump> is displayed.	Void
ValueExists(string:name) Does a value of the given name exist in the viewstate?	Name (required) - The name of the value to check	Boolean

View API

Inside of a .CFM template used as a view, two CFC instances are available in the unnamed or "variables" scope: viewstate and viewcollection. Viewstate is used to access values from the viewstate. Viewcollection is used to access the rendered HTML of prior views.

ViewCollection

In a .CFM template used as a View in a Model-Glue Application, the "viewcollection" variable gives access to the rendered results of prior views. This can be used to "stack" views. The following example shows a simple event handler in which one view, dspBody.cfm, is used inside of another (dspTemplate.cfm):

```
<!-- In ModelGlue.xml -->
<event-handler name="viewstacking">
  <views>
    <include name="body" template="dspBody.cfm" />
    <include name="main" template="dspTemplate.cfm" />
  </views>
</event-handler>
<!-- dspBody.cfm --->
Hi, I'm dspBody.cfm!
<!-- dspTemplate.cfm --->
<h1>dspTemplate.cfm</h1>
<cfoutput>#viewcollection.getView("body")#</cfoutput>
```

Function	Arguments	Returns
ViewCollection.Exists(name:string) Does a view of the given name exist in the viewcollection?	Name (required) - The name of the view to check	Boolean
ViewCollection.GetView(name:string) Returns the rendered results of a prior rendered view whose NAME attribute is the name argument passed. If the view does not exist, returns an empty string.	Name (required) - The name of the view's content to retrieve	String

Viewstate

In a .CFM template used as a View in a Model-Glue Application, the "viewstate" variable gives access to form and url data, along with any values added in a controller function through the Event.SetValue() method of the Event API.

Function	Arguments	Returns
Viewstate.Exists(name:string) Does a value of the given name exist in the viewstate?	Name (required) - The name of the value to check	Boolean
Viewstate.GetAll() Returns the structure that contains all viewstate members by reference.	None	Struct
Viewstate.GetValue(name:string [, default:any]) Returns a value from the viewstate, such as a form or url variable.	Name (required) - The name of the value to retrieve Default (optional) - If the value does not exist, a default value to set into the viewstate and then return	Any

Download: <http://www.model-glue.com>

Official Documentation: <http://docs.model-glue.com/>

Support Group: <http://groups.google.com/group/model-glue>