

ColdBox: Using 'Persist'

Posted At : June 10, 2008 5:48PM | Posted By : Matt Quackenbush

Related Categories: ColdBox, ColdFusion

Back in the [ColdBox 2.5.1 release](#) , a new feature was added: persist. Think of it as a sort of "flash memory", allowing your application to "remember" event variables when moving the user to the next event. I was recently asked about this feature, so I figured I would make a quick post about it in an effort to help more people.

In your [event handlers \(controllers\)](#) , it is often necessary to send the request along to another event. ColdBox provides two methods for doing this in your handlers:

1. setNextEvent() - for use with standard URLs
2. setNextRoute() - for use with SES URLs

In the [docs](#) , you'll see that there is an *optional* argument for these methods named 'persist'. Providing a comma-delimited list of **variable names** to this argument will assure you that the variables are carried over to the next event. It's very simple...

```

.. // for standard URLs
}. setNextEvent(
}. event: "my.event" ,
}. persist: "variable1,variable2,variable3"
}. );
}. //
}. // for SES URLs
}. setNextRoute(
}. route: "my/event" ,
}. persist: "variable1,variable2,variable3"
.. );

```

In this example, variable1, variable2, and variable3 will still exist in my.event when the request gets there. Schweeeeeet!! But, there is one MAJOR point to make, so pay really close attention here...

*In order to persist variables between events, they ***must*** exist in the event object, or request collection.*

In other words, these are **not** var'd variables within the event method.

```

.. <cffunction name= "doSomething" >
}. <!-- set the method arguments --->
}. <cfargument name= "Event" type= "coldbox.system.beans.requestContext" />
}. <cfscript>
}. var foo = "bar" ;
}. setNextRoute(
}. route: "foo/bar2" ,
}. persist: "foo"
}. );
.. </cfscript>
}. </cffunction>

```

In the example above, we're telling setNextRoute() to persist an event variable named 'foo'. As you can see, 'foo' is a var'd variable within the event method. This example **will not** result in 'foo' existing when the request gets to the next event.

The 'persist' argument is valid for *event variables* only, as shown in the next example.

```
.. <cffunction name= "doSomething" >
?. <!--- set the method arguments --->
?. <cfargument name= "Event" type= "coldbox.system.beans.requestContext" />
?. <cfscript>
?. var foo = "bar" ;
?. event.setValue( "foo" , foo);
?. setNextRoute(
?. route: "foo/bar2" ,
?. persist: "foo"
.. );
?. </cfscript>
?. </cffunction>
```

As you can see in this example, 'foo' exists in the event itself, so it **will** be passed along to the next event.

Just wanted to pass that along, since it seems to be a common misconception.