

# Zend\_Controller\_Action\_Helper\_AjaxContext

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

## Zend Framework: Zend\_Controller\_Action\_Helper\_AjaxContext Component Proposal

<b>Proposed Component Name</b>	Zend_Controller_Action_Helper_AjaxContext
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Controller_Action_Helper_AjaxContext">http://framework.zend.com/wiki/display/ZFDEV/Zend_Controller_Action_Helper_AjaxContext</a>
<b>Proposers</b>	Matthew Weier O'Phinney
<b>Revision</b>	0.8.0 - 03 January 2008: Initial proposal 0.9.0 - 04 January 2008: Updated proposal based on user feedback 0.9.1 - 04 January 2008: Added ContextSwitch helper to proposal; refactored to allow setting arbitrary context types 0.9.2 - 04 January 2008: Updated use cases to follow current laboratory code 0.9.3 - 06 January 2008: Updated use cases and skeletons to show getCurrentContext() method 0.9.4 - 07 January 2008: Updated UC-01 per Darby's comments 0.9.5 - 08 January 2008: Updated with feedback from Zend team; documents context enabling, adds support for building action controller context arrays, adds support for multiple headers, updates requirements to specify that the context will be prepended to existing view suffix for context suffix (wiki revision: 18)

## Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

### 1. Overview

Zend\_Controller\_Action\_Helper\_AjaxContext enables view context switching and sets appropriate response headers when XmlHttpRequests are detected.

### 2. References

- None

### 3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** be optionally loaded, on-demand, as an action helper.
- This component **will** disable layouts when XHR requests are detected.
- This component **will** use properties from the current action controller and request object to determine if context switching should occur.
- This component **will** utilize view script suffixes and HTTP response headers to accomplish context switching.
- This component **will** default to HTML responses unless otherwise requested.
- This component **will** use a **configurable** request parameter to determine the response context to utilize.
- This component **will** allow configuration of the headers to send for the contexts supported.
  - This component **will** allow multiple configurable headers per context
- This component **will** allow configuring the view suffix to use when context switching.
  - This component **will** simply prepend the suffix previously configured in the ViewRenderer with the new format.
  - This component **will** provide helper methods for creating and configuring the action controller context array

### 4. Dependencies on Other Framework Components

- Zend\_Controller\_Action\_Helper\_Abstract
- Zend\_Controller\_Action\_Helper\_ContextSwitch (part of this proposal)
- Zend\_Controller\_Action\_Helper\_ViewRenderer
- Zend\_Controller\_Request\_Http
- Zend\_Controller\_Response\_Http
- Zend\_Layout

### 5. Theory of Operation

Zend\_Controller\_Action\_Helper\_AjaxContext is designed to make detecting XMLHttpRequests and switching view context trivial. When called, it checks to see if the request is an XHR request; if so, it then disables layouts. A check is made on the 'ajaxable' property of the associated action controller, and if the current action is in that list, it then determines the response format requested; the viewRenderer is then updated with a new view script suffix representing the format, and appropriate response headers are set.

View scripts for handling AJAX contexts still end in the default view suffix, but have an additional suffix prepended. For instance, if the ViewRenderer's default suffix is '.phtml', and an XML context is selected, the new suffix will be 'xml.phtml'; the JSON context becomes 'json.phtml', and HTML becomes 'ajax.phtml' (to differentiate from the standard HTML returned by the action).

AjaxContext would itself extend another helper, ContextSwitch. This does the heavy lifting; AjaxContext itself only performs the isXmlHttpRequest() check. It also allows specifying arbitrary contexts via accessors.

To select a context, you simply add an extra parameter to the request. By default, this parameter is 'format'. As an example, if you were visiting the path /foo/bar and wanted the XML context, you could specify /foo/bar?format=xml, or, using path parameters as supported by the default router, /foo/bar/format/xml.

### 6. Milestones / Tasks

- Milestone 1: [DONE] Initial proposal published for review
- Milestone 2: [DONE] Tests and initial code checked into the laboratory
- Milestone 3: Final code with >80% coverage checked into incubator
- Milestone 4: Documentation completed
- Milestone 5: Promotion to core

### 7. Class Index

- Zend\_Controller\_Action\_Helper\_ContextSwitch

- Zend\_Controller\_Action\_Helper\_AjaxContext

## 8. Use Cases

## 9. Class Skeletons

```
]]></ac:plain-text-body></ac:macro>  
]]></ac:plain-text-body></ac:macro>
```