

TYPO3.Flow - Bug # 2580

Status:	Resolved	Priority:	Must have
Author:	Malte Jansen	Category:	MVC
Created:	2009-02-05	Assigned To:	
Updated:	2010-10-20	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Affected Flow version:			
Subject:	Crash in MVC-Controller		
Description			
<p>The function initializeAction() of the ActionController does not have the access to the to \$this->view, because the first the function is called the view is not initialized.</p> <p>The function is called twice</p> <ol style="list-style-type: none">1. while creating the object (after the injections) --> \$this->view is empty2. while callActionMethod() --> the view is initialized <p><i>Expected result:</i></p> <p>The view is available in initializeAction().</p> <p><i>Solution:</i></p> <p>Rename initializeAction() to preAction().</p> <p>Perhaps some postAction() would also be pretty useful.</p>			

Associated revisions

Revision 095abbb8 - 2009-02-09 16:01 - Robert Lemke

Known issue in this commit: the TYPO3 backend and TYPO3CR backend don't work at the moment because the REST controller is not fully functional.

- FLOW3: (MVC) The canProcessRequest() method in the ActionController now also accepts sub classes of the supported request types. Fixes #2601
- FLOW3: (MVC) Refactored parts of the ActionController. Also changed the order in which certain methods are called. The order is now: resolveActionMethodName(), initializeArguments(), mapRequestArgumentsToLocalArguments(), initializeView(), initializeAction(), callActionMethod(). Also resolves #2580.
- FLOW3: (MVC) Fixed parts of the RESTController. Note: the show, delete and update actions currently don't work due to a new way of handling ids. I'm working on it ...
- FLOW3: (MVC) Wrote proper tests for the REST controller. There are still missing tests for resolveActionMethodName() method.

History

#1 - 2009-02-05 20:12 - Karsten Dambekalns

The first call should be to initializeObject(), it is supposed to be called right after any injections have been done...

My gut feeling tells me that in initializeAction() the view should be available, indeed.

#2 - 2009-02-05 23:53 - Malte Jansen

Have I read it later.

Perhaps there just must be two lines switched...

ActionController:

```
protected function callActionMethod() {
    $actionMethodName = $this->request->getControllerActionName() . 'Action';
    if (!method_exists($this, $actionMethodName)) throw new \F3\FLOW3\MVC\Exception\NoSuchAction('An action "' .
$this->request->getControllerActionName() . '" does not exist in controller "' . get_class($this) . '".', 1186669086);
    $this->initializeAction();
    if ($this->initializeView) $this->initializeView();
    $actionResult = call_user_func_array(array($this, $actionMethodName), array());
    if (is_string($actionResult) && strlen($actionResult) > 0) {
        $this->response->appendContent($actionResult);
    }
}
```

Just call the initializeView() before initializeAction().

Haven't check it, but if the view is not init, than it should not be available in the initAction...

#3 - 2009-02-06 00:01 - Malte Jansen

By the way in the AbstractController the function initializeArguments() should be protected like all others init-functions.

#4 - 2009-02-09 17:03 - Robert Lemke

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset r1884.