

TYPO3.Flow - Feature # 38222

Status:	New	Priority:	Could have
Author:	Adrian Föder	Category:	Core
Created:	2012-06-20	Assigned To:	
Updated:	2013-02-21	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Step execution signals with concrete name		
Description			
<p>currently, Core\Booting\Sequence::invokeStep() looks</p> <pre>1\$bootstrap->getSignalSlotDispatcher()->dispatch(__CLASS__, 'beforeInvokeStep', array(\$step)); 2\$identifier = \$step->getIdentifier(); 3\$step(\$bootstrap); 4\$bootstrap->getSignalSlotDispatcher()->dispatch(__CLASS__, 'afterInvokeStep', array(\$step));</pre>			
<p>This requires listening to a specific step invocation to look like this (taken from an example):</p> <pre>1\$dispatcher->connect('TYPO3\FLOW3\Core\Booting\Sequence', 'afterInvokeStep', function(\TYPO3\FLOW3\Core\Booting\ Step \$step) use (\$bootstrap) { 2 if (\$step->getIdentifier() === 'typo3.flow3:objectmanagement:compiletime:create') { 3 \$bootstrap->getObjectManager()->get('....'); 4 } 5});</pre>			
<p>Due to the fact that, I claim, invoke-step-listening surely always intends to affect a specific step, I suggest to involve additional "magic" signals that take their name from the step identifier, resulting in, for example,</p> <pre>1\$dispatcher->connect('TYPO3\FLOW3\Core\Booting\Sequence', ' afterInvokeStepTypo3Flow3ObjectmanagementCompiletimeCreate', function(\TYPO3\FLOW3\Core\Booting\Step \$step) use (\$bootstrap) { 2 \$bootstrap->getObjectManager()->get('....'); 3});</pre>			

History

#1 - 2013-02-21 16:39 - Adrian Föder

- Assigned To deleted (Adrian Föder)