

TYPO3.Flow - Bug # 37473

Status:	New	Priority:	Must have
Author:	Adrian Föder	Category:	- Testing -
Created:	2012-05-24	Assigned To:	
Updated:	2012-10-25	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: Git master			
Subject:	Subsequent Exceptions related to Doctrine Entity Manager makes it snap shut		
Description			
<p>If any exception occurs while persisting, and a subsequent test also tries to persist, an ORM Exception occurs telling that the Entity manager were closed.</p> <p>Reproduction code in Functional Test:</p> <pre>1 /** 2 * Fine, this throws, as expected, an Exception due to validation problems 3 * 4 * @test 5 * @expectedException \TYPO3\FLOW3\Persistence\Exception\ObjectValidationFailedException 6 */ 7 public function firstFailingTestThatCorrectlyThrowsException() { 8 \$invalidAccount = new \TYPO3\FLOW3\Security\Account(); 9 \$this->persistenceManager->add(\$invalidAccount); 10 \$this->persistenceManager->persistAll(); 11 } 12 13 /** 14 * This (same code) throws now "Doctrine\ORM\ORMException: The EntityManager is closed." 15 * 16 * @test 17 */ 18 public function secondFailingTestThatSaysEntityManagerWasClosed() { 19 \$invalidAccount = new \TYPO3\FLOW3\Security\Account(); 20 \$this->persistenceManager->add(\$invalidAccount); 21 \$this->persistenceManager->persistAll(); 22 } 23</pre>			
<p>Maybe this is some kind related:</p> <p>http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/transactions-and-concurrency.html#exception-handling</p>			

History

#1 - 2012-10-25 10:29 - Matthieu Napoli

Apparently this is the intended behavior of Doctrine:

In other words, when an exception occurs during flush/commit it is too error-prone to try to get the UoW back into a reusable state. Such an

exception is unrecoverable for the EM/UoW, hence why it is closed and should not be reused.

[...]

You should not reuse an EM in tests like that as it causes too many potential side-effects between tests, which should be avoided. The Doctrine2 tests themselves get a fresh EM for every test method that needs one. What is reused/shared is the database connection in the functional tests. Creating a new EM is not a costly operation. A newly created EM can use the same DB connection (and Configuration and EventManager) as a previously failed one.

[...]

Your code can recover from the exception, if you so desire, but the EM/UoW in which this exception occurred can not recover. Thus, if **you** can recover and you need a new unit of work, you should create one (by creating a new EM).

from https://groups.google.com/forum/?fromgroups=#topic/doctrine-user/cE7h5Mud_hc