

TYPO3.Flow - Bug # 32415

Status:	Resolved	Priority:	Must have
Author:	Fernando Arconada	Category:	
Created:	2011-12-09	Assigned To:	Karsten Dambekalns
Updated:	2012-01-13	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: Git master			
Subject:	Authentication + Doctrine event listeners = apache segfault		
Description			
here it is the proof of concept https://github.com/farconada/F2.PoC			
If you try to login with a valid username password the apache segfaults If you disable <code>\$this->persistenceManager->injectEntityManager(\$entityManager);</code> the all works ideas?			

History

#1 - 2011-12-19 13:43 - Karsten Dambekalns

- Status changed from New to Accepted
- Assigned To set to Karsten Dambekalns
- Target version set to 1.1

#2 - 2011-12-19 14:24 - Karsten Dambekalns

- Status changed from Accepted to Needs Feedback

I had feared that, and I was right: your code works fine on my machine (after fixing the one missing class name change for the listener interface).

Segmentation faults are a classic and when PHP segfaults you can sometimes change your code to fix it. But more often than not I have seen segfaults go away or occur only with specific combinations of libraries used by PHP, compile-time option changes or extensions being added or removed.

I don't really know how to proceed here, even if I replicated your setup and could reproduce the error, it might just as well not be fixable.

#3 - 2011-12-19 14:52 - Fernando Arconada

Tested in another machine, and it crash also.

Tested with Fedora 15 32bit and Fedora 15 64bit

I could try with CentOS 5.7 also.

Any idea, time ago I had another segfault with was fixed in FLOW3 core, was related to AOP, no idea in this situation.

I could also give you access to a computer that segfaults.

What do you think

PD: it only crash after successfully login with username/password and default provider in developer environment

#4 - 2011-12-19 15:04 - Fernando Arconada

the core dump backtrace

```
[nemesisis@root ~]# gdb /usr/sbin/httpd /tmp/core.30141
GNU gdb (GDB) Fedora (7.3.1-46.fc15)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/&gt;...
Reading symbols from /usr/sbin/httpd...(no debugging symbols found)...done.
[New LWP 30141]
[Thread debugging using libthread_db enabled]
Core was generated by `usr/sbin/httpd'.
Program terminated with signal 11, Segmentation fault.
#0 0x01290a50 in zend_object_store_get_object () from /etc/httpd/modules/libphp5.so
Missing separate debuginfos, use: debuginfo-install httpd-2.2.21-1.fc15.i686
(gdb) bt
#0 0x01290a50 in zend_object_store_get_object () from /etc/httpd/modules/libphp5.so
#1 0x0128c5bc in zend_objects_get_address () from /etc/httpd/modules/libphp5.so
#2 0x0128c7ec in zend_std_get_properties () from /etc/httpd/modules/libphp5.so
#3 0x00e8dcbc in igbinary_serialize_array_sleep (incomplete_class=false, h=0x23d78f78, ce=0x23bb9fcc,
z=0x23c78dec, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1011
#4 igbinary_serialize_object (z=0x23c78dec, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1187
#5 igbinary_serialize_zval (z=<optimized out>, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1223
#6 igbinary_serialize_zval (igsd=0xbfb20d30, z=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:1215
#7 0x00e8ddc7 in igbinary_serialize_array_sleep (incomplete_class=false, h=0x23d70450, ce=0x239a8dd4,
z=0x23856690, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1037
#8 igbinary_serialize_object (z=0x23856690, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1187
#9 igbinary_serialize_zval (z=<optimized out>, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1223
#10 igbinary_serialize_zval (igsd=0xbfb20d30, z=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:1215
#11 0x00e90516 in igbinary_serialize_array_sleep (incomplete_class=false, h=0x23d6ffd8, ce=0x22d06008,
z=0x22d18d98, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1037
#12 igbinary_serialize_object (z=0x22d18d98, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1187
#13 igbinary_serialize_zval (igsd=0xbfb20d30, z=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:1223
#14 0x00e8b8fa in igbinary_serialize8 (i=<optimized out>, igsd=<optimized out>)
at /root/igbinary-1.1.1/igbinary.c:627
#15 igbinary_serialize8 (i=<optimized out>, igsd=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:1215
#16 igbinary_serialize_null (igsd=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:681
#17 igbinary_serialize_zval (z=0x22d18d98, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1221
#18 igbinary_serialize_array (incomplete_class=false, object=false, z=<optimized out>, igsd=0xbfb20d30)
at /root/igbinary-1.1.1/igbinary.c:895
#19 igbinary_serialize_zval (igsd=0xbfb20d30, z=<optimized out>) at /root/igbinary-1.1.1/igbinary.c:1225
#20 0x00e892b5 in igbinary_serialize_array_sleep (incomplete_class=false, h=0x23d720a8, ce=0x22cb70ec,
z=0x22cbb380, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1037
#21 igbinary_serialize_object (z=0x22cbb380, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1187
```

```
#22 igbinary_serialize_zval (z=<optimized out>, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1223
#23 igbinary_serialize_zval (z=<optimized out>, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:818
#24 igbinary_serialize_array (igsd=0xbfb20d30, z=<optimized out>, object=false, incomplete_class=false)
  at /root/igbinary-1.1.1/igbinary.c:895
#25 0x00e97eb1 in igbinary_serialize_zval (z=0x23d725a0, igsd=0xbfb20d30) at /root/igbinary-1.1.1/igbinary.c:1225
#26 igbinary_serialize_array (incomplete_class=false, object=false, z=<optimized out>, igsd=0xbfb20d30)
  at /root/igbinary-1.1.1/igbinary.c:895
#27 ps_srlzr_encode_igbinary (newstr=0xbfb20e0c, newlen=0xbfb20efc) at /root/igbinary-1.1.1/igbinary.c:447
#28 0x011523fd in ?? () from /etc/httpd/modules/libphp5.so
#29 0x01154d0d in ?? () from /etc/httpd/modules/libphp5.so
---Type <return> to continue, or q <return> to quit---
#30 0x012fd606 in execute_internal () from /etc/httpd/modules/libphp5.so
#31 0x00a9b301 in xdebug_execute_internal () from /usr/lib/php/modules/xdebug.so
#32 0x012e1365 in ?? () from /etc/httpd/modules/libphp5.so
#33 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#34 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#35 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#36 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#37 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#38 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#39 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#40 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#41 0x0125e2a3 in zend_call_function () from /etc/httpd/modules/libphp5.so
#42 0x011a53bc in ?? () from /etc/httpd/modules/libphp5.so
#43 0x012fd606 in execute_internal () from /etc/httpd/modules/libphp5.so
#44 0x00a9b301 in xdebug_execute_internal () from /usr/lib/php/modules/xdebug.so
#45 0x012e1365 in ?? () from /etc/httpd/modules/libphp5.so
#46 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#47 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#48 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#49 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#50 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#51 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#52 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#53 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#54 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#55 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#56 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#57 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#58 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#59 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#60 0x012e1061 in ?? () from /etc/httpd/modules/libphp5.so
#61 0x01291a0e in execute () from /etc/httpd/modules/libphp5.so
#62 0x00a9b00a in xdebug_execute () from /usr/lib/php/modules/xdebug.so
#63 0x0126c9c9 in zend_execute_scripts () from /etc/httpd/modules/libphp5.so
#64 0x01213370 in php_execute_script () from /etc/httpd/modules/libphp5.so
#65 0x012fee30 in ?? () from /etc/httpd/modules/libphp5.so
#66 0x009c992e in ap_run_handler ()
#67 0x009c9da6 in ap_invoke_handler ()
---Type <return> to continue, or q <return> to quit---
#68 0x009d839c in ap_internal_redirect ()
#69 0x0040fcf2 in ?? () from /etc/httpd/modules/mod_rewrite.so
#70 0x009c992e in ap_run_handler ()
```

#71 0x009c9da6 in ap_invoke_handler ()
#72 0x009d8d20 in ap_process_request ()
#73 0x009d5850 in ?? ()
#74 0x009d0eee in ap_run_process_connection ()
#75 0x009de57d in ?? ()
#76 0x009deeeb in ?? ()
#77 0x009defbc in ?? ()
#78 0x009dfba0 in ap_mpm_run ()
#79 0x009b272f in main ()

#5 - 2011-12-19 23:26 - Karsten Dambekalns

What happens if you disable igbinary?

#6 - 2011-12-19 23:46 - Fernando Arconada

I have disabled some php extensions but nothing

I have updated the package at GitHub cause I have a clue: Only segfaults if you try to login with a User created with the Command inside the package

#7 - 2012-01-04 20:42 - Karsten Dambekalns

- *Status changed from Needs Feedback to Accepted*

I can reproduce the segmentation fault with the updated PoC code.

It occurs with current master and is not fixed with the updated Doctrine 2 libraries.

#8 - 2012-01-05 16:19 - Karsten Dambekalns

- *% Done changed from 0 to 100*

Ok, I finally wanted to see if this goes away when not adding any event listeners. So I checked out the AbstractController. Aha. You create a new EntityManager and inject it to the PersistenceManager. **Do not do that.**

A fix is available in <https://github.com/farconada/F2.PoC/pull/1>

#9 - 2012-01-05 16:19 - Karsten Dambekalns

- *Project changed from TYPO3 Flow Base Distribution to TYPO3.Flow*

#10 - 2012-01-05 16:26 - Karsten Dambekalns

- *Status changed from Accepted to Needs Feedback*

- *Has patch set to No*

#11 - 2012-01-06 13:52 - Fernando Arconada

I have tested your solution and works quite well

So the best way to use doctrine event listeners is:

```
//Event listener para indexacion de objetos
$entityManager = $this->objectManager->get('Doctrine\Common\Persistence\ObjectManager');
$entityManager->getEventManager()->addEventListener(
    array(\Doctrine\ORM\Events::postUpdate, \Doctrine\ORM\Events::postPersist, \Doctrine\ORM\Events::preRemove),
    $this->doctrineEventListener
);
```

where `$this->doctrineEventListener` is a class that has methods like

```
public function postPersist(\Doctrine\ORM\Event\LifecycleEventArgs $args);
public function postUpdate(\Doctrine\ORM\Event\LifecycleEventArgs $args);
public function preRemove(\Doctrine\ORM\Event\LifecycleEventArgs $args);
```

Thanks for your solution

#12 - 2012-01-13 17:01 - Karsten Dambekalns

- *Status changed from Needs Feedback to Resolved*