

TYPO3.Flow - Bug # 52907

Status:	Resolved	Priority:	Should have
Author:	Kay Strobach	Category:	Persistence
Created:	2013-10-17	Assigned To:	Bastian Waidelich
Updated:	2014-03-20	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: Flow 2.0.0			
Subject:	#1355480641: Uncaught Exception		
Description			
<p>OneToMany mapping on field 'measurementDataRecords' requires the 'mappedBy' attribute.</p> <p>More Information Exception code #0 File /project/Packages/Libraries/doctrine/orm/lib/Doctrine/ORM/Mapping/MappingException.php line 150</p> <p>The Exception message should name the declaring class for the not interpretable property ...</p>			

Associated revisions

Revision 7e0c425c - 2013-10-17 12:02 - Bastian Waidelich

[BUGFIX] Expose more details in MappingExceptions

This adds a try/catch block to the FlowAnnotationDriver adding the affected class name to the message of the thrown exception.

Before if ``evaluatePropertyAnnotations`` threw an exception during the ``flow:doctrine:compileproxies`` command the exception did not contain details about the affected class (see #52907).

Change-Id: I6933c1bf7c206a9d2600024686e3bdfcff5adf06

Fixes: #52907

Releases: master, 2.0

Revision 86c578af - 2013-12-02 13:03 - Bastian Waidelich

[BUGFIX] Expose more details in MappingExceptions

This adds a try/catch block to the FlowAnnotationDriver adding the affected class name to the message of the thrown exception.

Before if ``evaluatePropertyAnnotations`` threw an exception during the ``flow:doctrine:compileproxies`` command the exception did not contain details about the affected class (see #52907).

Change-Id: I6933c1bf7c206a9d2600024686e3bdfcff5adf06

Fixes: #52907

Releases: master, 2.0

History

#1 - 2013-10-17 09:54 - Adrian Förder

- Status changed from *New* to *Rejected*

Hello Kay,

this is a Doctrine exception, hence out of the competence of Flow...

Feel free to reopen if you disagree in any kind.

Best,

Adrian

#2 - 2013-10-17 10:10 - Kay Strobach

mhmm ok, but it would be possible to catch and enhance it ;) - for usability reasons a good error message would be great ;)

#3 - 2013-10-17 10:15 - Adrian Förder

I think that would open kind of a "box" if it's started to catch these kind of errors. I even suspect that's rather hard because Flow only "prepares" the data Doctrine is fed with, and hands it over completely to doctrine later than.

#4 - 2013-10-17 10:21 - Bastian Waidelich

- File *exception.png* added

- Status changed from *Rejected* to *Needs Feedback*

Hi Kay,

can you give some more context? Where do you get this exception? If I add a property:

```
1 /**
2  * @var \Doctrine\Common\Collections\ArrayCollection<\MyPackage\Domain\Model\Product>
3  * @ORM\OneToMany()
4  */
5 protected $foos;
```

I get a wrapping exception that shows the affected class name: *exception.png*

#5 - 2013-10-17 10:22 - Bastian Waidelich

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

- Assigned To set to Bastian Waidelich

#6 - 2013-10-17 10:27 - Kay Strobach

keep it closed - i agree. I had a misconfigured Entity (created out of the context and then copied into a FLOW project)
If normally there is a wrapping exception this is totally fine.

#7 - 2013-10-17 10:31 - Bastian Waidelich

- Has patch set to No

Kay Strobach wrote:

```
| keep it closed - i agree. I had a misconfigured Entity (created out of the context and then copied into a FLOW project)
| If normally there is a wrapping exception this is totally fine.
```

Sorry, what I meant with "context" is:

- What did you change? (you created/pasted a misconfigured entity to Flow, right?)
- When did the exception occur?

or other words: How can we reproduce this? ;)

#8 - 2013-10-17 10:46 - Kay Strobach

this is how i produced the exception.

2013-10-17_10-41-58.png

```
**
* set of parameters for an abstract measurement series
*
* @Flow\Entity
*
*/
class MeasurementSeries {

    /**
     * @ORM\OneToMany
     * @var \Doctrine\Common\Collections\Collection<\Vendor\Logger\Domain\Model\MeasurementDataRecord>
     */
    protected $measurementDataRecords = null;
}
```

#9 - 2013-10-17 10:47 - Kay Strobach

- File 2013-10-17_10-41-58.png added

#10 - 2013-10-17 11:57 - Bastian Waidelich

- *File exception_new.png added*
- *Category set to Persistence*
- *Status changed from Needs Feedback to Accepted*
- *Target version deleted (2.0.1)*

Ok, thanks to your feedback I could reproduce the issue (it happens only after flushing the cache, after a refresh you get the nested exception).
I could wrap the code that throws the exception so that the initial exception now contains more details:
exception_new.png

I'm pushing review request now, hoping for feedback from Karsten (aka Mr. Doctrine)

#11 - 2013-10-17 12:03 - Gerrit Code Review

- *Status changed from Accepted to Under Review*

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24862>

#12 - 2013-10-17 12:27 - Kay Strobach

THX

#13 - 2013-10-17 14:36 - Bastian Waidelich

- *Status changed from Under Review to Resolved*
- *% Done changed from 0 to 100*

Applied in changeset commit:7e0c425ceefd06252ef00b9addcd9389f1e7d950.

#14 - 2013-12-02 13:03 - Gerrit Code Review

- *Status changed from Resolved to Under Review*

Patch set 1 for branch **2.0** of project **Packages/TYPO3.Flow** has been pushed to the review server.

It is available at <https://review.typo3.org/25873>

#15 - 2014-03-20 19:59 - Kay Strobach

can be closed imho, as gerrit states it as resolved ...

#16 - 2014-03-20 20:37 - Bastian Waidelich

- *Status changed from Under Review to Resolved*

Thx ;)

Files

exception.png	46.7 kB	2013-10-17	Bastian Waidelich
2013-10-17_10-41-58.png	145.1 kB	2013-10-17	Kay Strobach
exception_new.png	30.9 kB	2013-10-17	Bastian Waidelich