

TYPO3.Flow - Bug # 58153

Status:	New	Priority:	Should have
Author:	Ferdinand Kuhl	Category:	AOP
Created:	2014-04-24	Assigned To:	
Updated:	2014-04-24	Due date:	
PHP Version:	5.4		
Has patch:	No		
Complexity:			
Affected Flow version:	Flow 2.0.0		
Subject:	Session - Scope, Property with interface annotation fails at wakeup		
Description	<p>Assume I have an Object called Vendor\Package\Service\PackageSession which has Session scope. This object has a property, which is annotated with an interface, which some of my domain objects may implement.</p> <p>If this object is reconstituted the AOP Proxy crashes during wakeup with the following trace:</p> <pre> Uncaught exception in line 40 of Packages/Libraries/doctrine/common/lib/Doctrine/Common/Persistence/Mapping/RuntimeReflectionService.php: Class 'Vendor\Package\Domain\Model\EntityInterface' does not exist 40 Doctrine\Common\Persistence\Mapping\MappingException::nonExistingClass("Vendor\Package\Domain\Model\EntityInterface") ") 39 Doctrine\Common\Persistence\Mapping/RuntimeReflectionService::getParentClasses("Vendor\Package\Domain\Model\EntityInterface") nterface") 38 Doctrine\Common\Persistence\Mapping\AbstractClassMetadataFactory::getParentClasses("Vendor\Package\Domain\Model\EntityInterface ntityInterface") 37 Doctrine\Common\Persistence\Mapping\AbstractClassMetadataFactory::loadMetadata("Vendor\Package\Domain\Model\EntityInterface") Interface") 36 Doctrine\Common\Persistence\Mapping\AbstractClassMetadataFactory::getMetadataFor("Vendor\Package\Domain\Model\EntityInterface" ityInterface") 35 Doctrine\ORM\EntityManager::getReference("Vendor\Package\Domain\Model\EntityInterface", "072e6c49-6a9a-a791-bbc3-91a502d73e83") 34 TYPO3\Flow\Persistence\Doctrine\PersistenceManager_Original::getObjectByIdentifier("072e6c49-6a9a-a791-bbc3-91a502d73e83", "Ven 3e83", "Vendor\Package\Domain\Model\EntityInterface", TRUE) 33 TYPO3\Flow\Persistence\Doctrine\PersistenceManager::getObjectByIdentifier("072e6c49-6a9a-a791-bbc3-91a502d73e83", "Vendor\Package\Domain\Model\EntityInterface", TRUE) 32 call_user_func_array(array 2 , array 3) 31 TYPO3\Flow\Persistence\Doctrine\PersistenceManager::Flow_Aop_Proxy_invokeJoinPoint(TYPO3\Flow\Aop\JoinPoint) 30 TYPO3\Flow\Aop\Advice\AdviceChain::proceed(TYPO3\Flow\Aop\JoinPoint) 29 TYPO3\Flow\Security\Aspect\PersistenceQueryRewritingAspect_Original::checkAccessAfterFetchingAnObjectByIdentifier(TYPO3\Flow\Ac O3\Flow\Aop\JoinPoint) 28 TYPO3\Flow\Aop\Advice\AroundAdvice::invoke(TYPO3\Flow\Aop\JoinPoint) </pre>		

```

27 TYPO3\Flow\Aop\Advice\AdviceChain::proceed(TYPO3\Flow\Aop\JoinPoint)
26
TYPO3\Flow\Persistence\Doctrine\PersistenceManager::getObjectByIdentifier("072e6c49-6a9a-a791-bbc3-91a502d73e83",
"Vendor\Package\Domain\Model\EntityInterface", TRUE)
25 Vendor\Package\Service\PackageSession::__wakeup()
[...]
```

The Problem is that the proxy-class builder generates the following code:

```

<?php
if ($this->$propertyName instanceof \TYPO3\Flow\Persistence\Aspect\PersistenceMagicInterface &&
!\TYPO3\Flow\Core\Bootstrap::$staticObjectManager->get('TYPO3\Flow\Persistence\PersistenceManagerInterface')->isNewObject($this->
bject($this->$propertyName) || $this->$propertyName instanceof \Doctrine\ORM\Proxy\Proxy) {
    if (!property_exists($this, 'Flow_Persistence_RelatedEntities') || !is_array($this->Flow_Persistence_RelatedEntities)) {
        $this->Flow_Persistence_RelatedEntities = array();
        $this->Flow_Object_PropertiesToSerialize[] = 'Flow_Persistence_RelatedEntities';
    }
    $identifier =
\TYPO3\Flow\Core\Bootstrap::$staticObjectManager->get('TYPO3\Flow\Persistence\PersistenceManagerInterface')->getIdentifierByObject($this->
fierByObject($this->$propertyName);
    if (!$identifier && $this->$propertyName instanceof \Doctrine\ORM\Proxy\Proxy) {
        $identifier = current(\TYPO3\Flow\Reflection\ObjectAccess::getProperty($this->$propertyName, '_identifier', TRUE));
    }
    $this->Flow_Persistence_RelatedEntities[$propertyName] = array(
        'propertyName' => $propertyName,
        'entityType' => $className,
        'identifier' => $identifier
    );
    continue;
}
?>
```

Of course the interface is known to the ObjectManager, so the interface name stays in class name. Correct behaviour would be to ask the object for its class name. Interesting is, that if a doctrine-proxy is set this way, everything works as expected

History

#1 - 2014-04-24 12:46 - Ferdinand Kuhl

Sorry, the above snippet from ProxyBuilder is the wrong snippet.

The problem is here:

```

<?php
/**
 * Autogenerated Proxy Method
 */
public function __sleep() {
[...]
```

```

foreach ($allReflectedProperties as $reflectionProperty) {
    [...]
    if (is_object($this->$propertyName) && !$this->$propertyName instanceof \Doctrine\Common\Collections\Collection) {
        if ($this->$propertyName instanceof \Doctrine\ORM\Proxy\Proxy) {
            $className = get_parent_class($this->$propertyName);
        } else {
            $varTagValues = $reflectionService->getPropertyTagValues('Vendor\Package\Service\PackageSession', $propertyName, 'var');
            if (count($varTagValues) > 0) {
                $className = trim($varTagValues[0], '\\');
            }
            if (\TYPO3\Flow\Core\Bootstrap::$staticObjectManager->isRegistered($className) === FALSE) {
                $className =
                \TYPO3\Flow\Core\Bootstrap::$staticObjectManager->getObjectByNameByClassName(get_class($this->$propertyName));
            }
        }
        [...]
    }
    $this->Flow_Object_PropertiesToSerialize[] = $propertyName;
}
$result = $this->Flow_Object_PropertiesToSerialize;
return $result;
}

?>
</pre>

```