

TYPO3.Flow - Bug # 40449

Status:	Resolved	Priority:	Should have
Author:	Cedric Ziel	Category:	Validation
Created:	2012-08-30	Assigned To:	Karsten Dambekalns
Updated:	2013-08-14	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: FLOW3 1.1.0			
Subject:	DateTimeValidator doesn't work as expected		
Description			
<p>I have a fairly simple Model which has a property \$date:</p>			
<pre>/** * The actual date of the transaction * @var \DateTime * @FLOW3\Validate(type="DateTime") */ protected \$date;</pre>			
<p>My thoughts were to use the DateTime Validator bundled with TYPO3.FLOW3.</p>			
<p>If I submit the form and (of what I understood, the validators do some serious business..) I get an error:</p>			
<pre>Invalid validate annotation in Cziel\F3finances\Domain\Model\Transaction::date: Could not resolve class name for validator "DateTime"</pre>			
<p>Stacktrace:</p>			
<pre>TYPO3\FLOW3\Validation\ValidatorResolver_Original::buildBaseValidatorConjunction("Cziel\F3finances\Domain\Model\Transaction##Default##Controller", "Cziel\F3finances\Domain\Model\Transaction", array) 17 TYPO3\FLOW3\Validation\ValidatorResolver_Original::getBaseValidatorConjunction("Cziel\F3finances\Domain\Model\Transaction", array) 16 TYPO3\FLOW3\Mvc\Controller\ActionController_Original::initializeActionMethodValidators() 15 TYPO3\FLOW3\Mvc\Controller\ActionController_Original::processRequest(TYPO3\FLOW3\Mvc\ActionRequest, TYPO3\FLOW3\Http\Response) 14 TYPO3\FLOW3\Mvc\Dispatcher_Original::dispatch(TYPO3\FLOW3\Mvc\ActionRequest, TYPO3\FLOW3\Http\Response) 13 TYPO3\FLOW3\Mvc\Dispatcher::dispatch(TYPO3\FLOW3\Mvc\ActionRequest, TYPO3\FLOW3\Http\Response) 12 call_user_func_array(array, array) 11 TYPO3\FLOW3\Mvc\Dispatcher::FLOW3_Aop_Proxy_invokeJoinPoint(TYPO3\FLOW3\Aop\JoinPoint)</pre>			

```
10 TYPO3\FLOW3\Aop\Advice\AdviceChain::proceed(TYPO3\FLOW3\Aop\JoinPoint)

9
  TYPO3\FLOW3\Security\Aspect\RequestDispatchingAspect_Original::setAccessDeniedResponseHeader(TYPO3\FLOW3\Aop\JoinPoint)
  \JoinPoint)

8 TYPO3\FLOW3\Aop\Advice\AroundAdvice::invoke(TYPO3\FLOW3\Aop\JoinPoint)

7 TYPO3\FLOW3\Aop\Advice\AdviceChain::proceed(TYPO3\FLOW3\Aop\JoinPoint)

6
  TYPO3\FLOW3\Security\Aspect\RequestDispatchingAspect_Original::blockIllegalRequestsAndForwardToAuthenticationEntryPoints(TYPO3\FLOW3\Aop\JoinPoint)
  oints(TYPO3\FLOW3\Aop\JoinPoint)

5 TYPO3\FLOW3\Aop\Advice\AroundAdvice::invoke(TYPO3\FLOW3\Aop\JoinPoint)

4 TYPO3\FLOW3\Aop\Advice\AdviceChain::proceed(TYPO3\FLOW3\Aop\JoinPoint)

3 TYPO3\FLOW3\Mvc\Dispatcher::dispatch(TYPO3\FLOW3\Mvc\ActionRequest, TYPO3\FLOW3\Http\Response)

2 TYPO3\FLOW3\Http\RequestHandler::handleRequest()

1 TYPO3\FLOW3\Core\Bootstrap::run()
```

Associated revisions

Revision 0a4f1ca4 - 2013-01-14 18:56 - Karsten Dambekalns

[BUGFIX] Make use of DateTimeValidator possible

When one used Validate(type="DateTime") the ValidatorResolver would try to use DateTime as the validator itself. This change fixes that.

Change-Id: I5506e8f989b83b201a0c0e820261a9c97a3f514d

Fixes: #40449

Releases: master, 2.0, 1.1

Revision 6cdca9d9 - 2013-01-15 14:58 - Karsten Dambekalns

[BUGFIX] Make use of DateTimeValidator possible

When one used Validate(type="DateTime") the ValidatorResolver would try to use DateTime as the validator itself. This change fixes that.

Change-Id: I5506e8f989b83b201a0c0e820261a9c97a3f514d

Fixes: #40449

Releases: master, 2.0, 1.1

Revision 2cb49548 - 2013-03-20 09:55 - Karsten Dambekalns

[BUGFIX] Make use of DateTimeValidator possible

When one used Validate(type="DateTime") the ValidatorResolver would try to use DateTime as the validator itself. This change fixes that.

Change-Id: I5506e8f989b83b201a0c0e820261a9c97a3f514d

Fixes: #40449

Releases: master, 2.0, 1.1

History

#1 - 2012-08-30 21:17 - Cedric Ziel

I just wanted to add, that I *think*, I am using Validators the right way.. others do their job just as "NotEmpty".

If I missed something, please help me with more Information on this specific validator..

#2 - 2012-08-30 21:24 - Adrian Förder

- Category set to Validation

- Assigned To set to Adrian Förder

This is indeed strange, what you did Looks perfectly correct as far as I can see.

Are you sure your code snippet is exactly the one that FLOW3 is currently working on?

Do you have own validators running, and, just to go very sure; your source is complete and not modified in any way, especially regarding the DateTime validator?

What happens, for example, if you replace it with the DateTimeRange validator?

Thanks for your Feedback!

#3 - 2012-08-30 21:24 - Adrian Förder

- Status changed from New to Needs Feedback

#4 - 2012-08-30 21:30 - Cedric Ziel

I tested it with the DateTimeRange Validator; this one works.

#5 - 2012-09-02 12:22 - Cedric Ziel

Well, I verified this one according to your questions:

- Yes, this is the actual snippet, that is being processed.
- I have no own validators running, which could confuse the resolver
- No modifications
- The DateTimeRangeValidator is being found by the resolver, such as NotEmpty Validator for example.

#6 - 2012-09-03 15:51 - Adrian Förder

- Status changed from Needs Feedback to Accepted

Cedric, thanks for your information. I will check if I can reproduce it with a Functional Test, because I have no clue why this is...

#7 - 2012-09-12 10:01 - Karsten Dambekalns

- Target version set to 1.1.1

I think I know what is wrong... This is the code in resolveValidatorObjectName() inside ValidatorResolver. The first if is TRUE, as DateTime is known.

But it fails the check for implementing ValidatorInterface a few lines down...

```
if ($this->objectManager->isRegistered($validatorType)) {
    $possibleClassName = $validatorType;
} else {
    if (strpos($validatorType, ':') !== FALSE) {
        list($packageName, $packageValidatorType) = explode(':', $validatorType);
        $possibleClassName = sprintf('%s\Validation\Validator%sValidator', str_replace(':', '\\', $packageName),
        $this->getValidatorType($packageValidatorType));
    } else {
        $possibleClassName = sprintf('TYPO3\FLOW3\Validation\Validator%sValidator', $this->getValidatorType($validatorType));
    }
    if (!$this->objectManager->isRegistered($possibleClassName)) {
        return FALSE;
    }
}
```

Could you check what \$possibleClassName is found after this block of code?

#8 - 2012-12-14 15:09 - Karsten Dambekalns

- Target version changed from 1.1.1 to 2052

#9 - 2013-01-14 09:31 - Irene Höppner

The \$possibleClassName found is 'DateTime' instead of 'TYPO3\Flow\Validation\Validator\DateTimeValidator'. That means the first if statement of that block finds the php class 'DateTime' registered and thus doesn't do the resolving in the else part.

I'm going to create a test and a patch for that. Should I check for 'DateTime' explicitly or do have an idea for a more general approach?

#10 - 2013-01-14 12:25 - Karsten Dambekalns

- Status changed from Accepted to Under Review

- Assigned To changed from Adrian Förder to Karsten Dambekalns

See <https://review.typo3.org/17520>

#11 - 2013-01-15 14:37 - Karsten Dambekalns

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:0a4f1ca4260156fd981bf642048e3f825158887a.

#12 - 2013-03-20 09:55 - Gerrit Code Review

- *Status changed from Resolved to Under Review*

Patch set 2 for branch **FLOW3-1.1** has been pushed to the review server.

It is available at <https://review.typo3.org/17546>

#13 - 2013-08-02 11:22 - Karsten Dambekalns

- *Status changed from Under Review to Resolved*

#14 - 2013-08-14 15:39 - Karsten Dambekalns

- *Target version changed from 2052 to 1.1.1*