TYPO3.Flow - Bug # 7182

Status:	Resolved	**Priority:**	Should have
Author:	Felix Oertel	**Category:**	MVC
Created:	2010-04-08	**Assigned To:**	
Updated:	2012-03-14	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version:	FLOW3 1.0.0		
Subject:	Throw an exception if class for annotated dataType is missing		

Description

Hey,

if you have a typo in the annotation of an argument of your method, FLOW3 throws some kind of confusing exception. Let's assume we want to hand over an object and want to annotate the var F3\SomePackage\Model\Domain\SomeModel but we misspell and actually type F3\SomePackage\Model\Typo\SomeModel. What FLOW3 throws is:

#1: Catchable Fatal Error: Argument 1 passed to F3\SomePackage\Controller\SomeController::someAction() must be an instance of F3\SomePackage\Model\Typo\SomeModel, array given in /some/fancy/path/flow3/Packages/Application/SomePackage/Classes/Controller/SomeController.php line someHighNumber

That's confusing, because it's no problem that the argument is an array but that the corresponding class is not loaded and the DataMapper is not able to map the argument to an object.

This is generated in F3\FLOW3\MVC\Controller\Argument.php:435 where the value of an argument is simply returned if the corresponding class does not exist. This avoides mapping the argument to an object and we end up with the wrong type but not revealing this to the user.

I think we should catch this !class_exists() and check, if the annotated datatype is user-/framework-defined (matching F3\-schema). If it is, we throw an exception that we were not able to map because of the missing file. If it is an primitive type (incl. validation-related types) it's ok to just return the value.

History

#1 - 2010-04-08 18:51 - Felix Oertel

- File flow3_7182_patch.diff added

I filed a bug report for this before in #7076 for extbase. But I agree with Jochen that we should solve this in FLOW3 first and then port it back.

Patch attached.

#2 - 2010-04-08 19:05 - Christopher Hlubek

- Assigned To set to Robert Lemke

@Robert: We talked about this issue in Lübeck already. It's hard to track this error and the framework should provide a more meaningful error here.

I walked through the code with Felix today and we thought about when to throw an exception for a missing class. It's not enough just to skip primitive types since the data type in the argument can be a virtual type (e.g. "Text"). But I'm not sure if it's okay to match every class in "F3" for a missing class exception.

#3 - 2010-07-09 14:48 - Robert Lemke

- Status changed from New to Needs Feedback

- Assigned To changed from Robert Lemke to Christopher Hlubek

- Target version set to 1.0 alpha 11

Christopher Hlubek wrote:

> *@Robert: We talked about this issue in Lübeck already. It's hard to track this error and the framework should provide a more meaningful error here.*
>
> *I walked through the code with Felix today and we thought about when to throw an exception for a missing class. It's not enough just to skip primitive types since the data type in the argument can be a virtual type (e.g. "Text"). But I'm not sure if it's okay to match every class in "F3" for a missing class exception.*

What do you mean with "virtual type" - a non-qualified class name?

It's really a bit tricky because we basically have two problems here:

1. the mapper should fail earlier and produce a proper error message that **an object of that name** (sic!) is not registered with FLOW3
2. the fatal error when calling the method with wrong arguments should be caught and transformed in to a more meaningful exception

#4 - 2010-08-18 18:08 - Karsten Dambekalns

Hi,

I'm puzzled by this:

Christopher Hlubek wrote:

> *It's not enough just to skip primitive types since the data type in the argument can be a virtual type (e.g. "Text").*

FLOW3 evaluates the @param annotation in this case, and there you must use valid types, i.e. primitives of fully qualified class names.

No?

#5 - 2010-08-19 11:07 - Karsten Dambekalns

- Status changed from Needs Feedback to Accepted

- Assigned To changed from Christopher Hlubek to Karsten Dambekalns

- Start date changed from 2010-04-08 to 2010-08-18

- Estimated time set to 1.00

#6 - 2010-08-19 11:33 - Karsten Dambekalns

- Assigned To deleted (Karsten Dambekalns)

- Target version deleted (1.0 alpha 11)

Karsten Dambekalns wrote:

> *FLOW3 evaluates the @param annotation in this case, and there you must use valid types, i.e. primitives of fully qualified class names.*

Wrong. Dang, the $dataType can indeed be something like "Text". So this needs more thought than sensible one day before a release...

Observation: If you check for no backslash in the type and return the value in those cases, you get a step further. For TYPO3 Phoenix this doesn't help, though, as you get an exception on a type that is an interface, not a class. If you then add interface_exists() to class_exists(), it complains that the value was "a simple type" - but it was a Page proxy already.

#7 - 2011-10-21 13:20 - Karsten Dambekalns

- Affected Flow version set to FLOW3 1.0.0

#8 - 2012-03-14 11:20 - Christian Müller

- Status changed from Accepted to Resolved

- Has patch set to No

The "new" PropertyMapper catches this error already and gives a resonable message.

Files

flow3_7182_patch.diff	2.6 kB	2010-04-08	Felix Oertel