

## TYPO3.Flow - Bug #3130

### ValidatorResolver does not work for float/numeric/mixed typed method arguments

2009-04-18 17:44 - Bastian Waidelich

<b>Status:</b>	Resolved	<b>Start date:</b>	2009-04-18
<b>Priority:</b>	Should have	<b>Due date:</b>	
<b>Assignee:</b>	Bastian Waidelich	<b>% Done:</b>	0%
<b>Category:</b>	Validation	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.0 alpha 1	<b>Complexity:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>			

#### Description

While implementing view helpers for Fluid I'm struggling with validation of the view helper arguments:

```
/**
 * @param float $value The value to format
 */
public function render($value) {
}
```

leads to the exception:

```
No validator found for argument name "value" with type "float"
```

The same for @param numeric and probably some more.

Additionally, we should be able to skip the "XSS-Filter" in some cases..:

```
<f3:form.textarea value="Some 'Test'" />
```

leads to the exception:

```
Validation for argument name "value" in view helper "F3\Fluid\ViewHelpers\Form\TextareaViewHelper"
FAILED. Expected type: "string"; Given: "string".
```

internally, the error message is

```
The given subject was not a valid text (contained XML tags). Got: "Some 'Test'"
```

Maybe

```
@param mixed
```

should be possible to allow to skip validation (Raw)..

#### Associated revisions

##### Revision ef50988e - 2009-04-21 22:16 - Bastian Waidelich

- FLOW3: (Validation) added more types to unifyDataType() in ValidationResolver. Additionally passed dataType through ucfirst() in order to find matching Validators for @param types (which are lower case by default) arguments of type "mixed" will be passed to RawValidator now - this might need to be revised, I'm committing this anyways as these changes are needed for the upcoming Fluid-commit. Relates to #3130

##### Revision 2171 - 2009-04-21 22:16 - Bastian Waidelich

- FLOW3: (Validation) added more types to unifyDataType() in ValidationResolver. Additionally passed dataType through ucfirst() in order to find matching Validators for @param types (which are lower case by default) arguments of type "mixed" will be passed to RawValidator now - this might need to be revised, I'm committing this anyways as these changes are needed for the upcoming Fluid-commit. Relates to #3130

#### Revision 6fd4ce49 - 2009-04-22 17:03 - Robert Lemke

- FLOW3: Raised required PHP version to 5.3.0RC1
- FLOW3: (Error) The \F3\FLOW3\Error\Error class now supports \_\_toString()
- FLOW3: (MVC) Finished implementation of validation rules in the action methods. Now these rules are actually enforced. Now really resolves #3092.
- FLOW3: (MVC) The ActionController now checks if it can process the request – didn't do that before. Fixes #3121.
- FLOW3: (Property) The property mapper's mapAndValidate() method now actually validates the source.
- FLOW3: (Reflection) The Reflection Service now detects class file changes again. (Problem was that the settings were not injected anymore because autowiring was turned off for the Reflection Service in the Objects.yaml) Fixes #3152.
- FLOW3: (Validation) Removed the Validation\Errors class - now a simple array is used instead.
- FLOW3: (Validation) Removed the \$errors argument from FilterInterface::filter(). More changes to this interface will follow.
- FLOW3: (Validation) Validation options must be set with the new setOptions() method instead of passing them to the isValid() method.
- FLOW3: (Validation) The errors array is not passed to the isValid() method of a validator anymore - instead it can be retrieved by a new getErrors() method which is defined in the ValidatorInterface.
- FLOW3: (Validation) Validators may now be singleton or prototype. Marked all those validators as prototype which require options and let others be singletons.
- FLOW3: (Validation) Removed the public resolveValidatorClassName() method from the ValidatorResolver class.
- FLOW3: (Validation) !!! changed the option names of the StringLength validator: now they are "minimum" and "maximum"
- FLOW3: (Validation) Changed the ObjectValidatorInterface: canValidateType() is now canValidate() and asks for an objects instead of a class name.
- FLOW3: (Validation) The first argument of the Validator Resolver's createMethodValidator() has changed: Now either the short name or a fully qualified object name of a *validator* is expected (instead of the data type)! Maybe also addresses #3130.
- FLOW3: (Documentation) Added a few lines of documentation and an example for the validation rules feature. More to come.
- Fluid: (Core) Fixed a test which was referring to the \F3\FLOW3\Validation\Errors class.

#### Revision 2175 - 2009-04-22 17:03 - Robert Lemke

- FLOW3: Raised required PHP version to 5.3.0RC1
- FLOW3: (Error) The \F3\FLOW3\Error\Error class now supports \_\_toString()
- FLOW3: (MVC) Finished implementation of validation rules in the action methods. Now these rules are actually enforced. Now really resolves #3092.
- FLOW3: (MVC) The ActionController now checks if it can process the request – didn't do that before. Fixes #3121.
- FLOW3: (Property) The property mapper's mapAndValidate() method now actually validates the source.
- FLOW3: (Reflection) The Reflection Service now detects class file changes again. (Problem was that the settings were not injected anymore because autowiring was turned off for the Reflection Service in the Objects.yaml) Fixes #3152.
- FLOW3: (Validation) Removed the Validation\Errors class - now a simple array is used instead.
- FLOW3: (Validation) Removed the \$errors argument from FilterInterface::filter(). More changes to this interface will follow.
- FLOW3: (Validation) Validation options must be set with the new setOptions() method instead of passing them to the isValid() method.
- FLOW3: (Validation) The errors array is not passed to the isValid() method of a validator anymore - instead it can be retrieved by a new getErrors() method which is defined in the ValidatorInterface.
- FLOW3: (Validation) Validators may now be singleton or prototype. Marked all those validators as prototype which require options and let others be singletons.
- FLOW3: (Validation) Removed the public resolveValidatorClassName() method from the ValidatorResolver class.
- FLOW3: (Validation) !!! changed the option names of the StringLength validator: now they are "minimum" and "maximum"
- FLOW3: (Validation) Changed the ObjectValidatorInterface: canValidateType() is now canValidate() and asks for an objects instead of a class name.
- FLOW3: (Validation) The first argument of the Validator Resolver's createMethodValidator() has changed: Now either the short name or a fully qualified object name of a *validator* is expected (instead of the data type)! Maybe also addresses #3130.
- FLOW3: (Documentation) Added a few lines of documentation and an example for the validation rules feature. More to come.
- Fluid: (Core) Fixed a test which was referring to the \F3\FLOW3\Validation\Errors class.

## History

---

### #1 - 2009-04-18 17:45 - Bastian Waidelich

- File FLOW3\_Validation\_3130\_01.patch added

### #2 - 2009-04-18 17:45 - Bastian Waidelich

- Assignee deleted (Andreas Förthner)

### #3 - 2009-04-22 09:12 - Bastian Waidelich

- Status changed from New to Needs Feedback

If committed a fix for this in r2171. This needs verification as I don't have an overview of all implications. ValidatorResolver probably needs some rework, but that might go into new tickets.

**#4 - 2009-05-19 12:34 - Robert Lemke**

- Target version set to 283

**#5 - 2009-06-04 09:53 - Bastian Waidelich**

- Status changed from Needs Feedback to Resolved

- Assignee set to Bastian Waidelich

- Target version changed from 283 to 1.0 alpha 1

I close this issue as a solution was introduced already in r2171. ValidationResolver probably needs some rework, but that should go into new tickets.

**Files**

---

FLOW3_Validation_3130_01.patch	509 Bytes	2009-04-18	Bastian Waidelich
--------------------------------	-----------	------------	-------------------