

TYPO3.Flow - Feature # 28399

Status:	Needs Feedback	Priority:	Should have
Author:	Christopher Hlubek	Category:	Validation
Created:	2011-07-22	Assigned To:	
Updated:	2012-12-10	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Validation message and code should be configurable for bundled validators		
Description			
<p>For custom applications the bundled validators of FLOW3 are nice but unusable if different messages and codes have to be used (e.g. for a custom Regex Validator).</p> <p>It would be more flexible to have two options code and message per Validator (or multiple if more than one message is added) that can be configured by the user. Additionally a message type ("error" or "warning") would be nice.</p> <p>The code change is pretty straight forward and should be compatible to the current version:</p> <pre>1 protected function isValid(\$value) { 2 ... 3 // If invalid 4 \$message = isset(\$this->options['message']) ? \$this->options['message'] : 'Default error message'; 5 \$code = isset(\$this->options['code']) ? \$this->options['code'] : 12345678; 6 \$this->addError(\$message, \$code); 7 }</pre>			
Related issues:			
related to TYPO3.Flow - Feature # 6456: Customization/localization of validat...		Closed	
related to TYPO3.Flow - Feature # 33970: Add translation files for bundled va...		Resolved	2012-02-16
related to TYPO3.Flow - Feature # 43841: Add package support to validation er...		New	2012-12-10

History

#1 - 2012-02-16 10:46 - Karsten Dambekalns

- Category set to Validation

- Has patch set to No

#2 - 2012-02-16 12:08 - Bastian Waidelich

Thinking about it - with the ability to use L18N for outputting the validation message, I don't think we need that feature anymore (see #33970) . So I'd suggest to close this.

If you think we do need to override the code / message I would suggest not to use "magic" options for this. Maybe we need a "ValidatorOptions" object anyways..

#3 - 2012-03-14 14:53 - Christian Müller

- Status changed from New to Rejected

With the current state of (validation) errors you can pretty much do all that in your output by using the code and arguments.

#4 - 2012-03-14 14:57 - Christopher Hlubek

- Status changed from Rejected to Needs Feedback

Well, actually it's not about the output. A validator is a prototype that is configured with options. To differentiate validation errors in a result object (take a regexp validator for example) it's sometimes necessary to get different codes depending on the configured instance of a validator (e.g. one code for invalid email and one for invalid zip code). This is especially interesting when building web services that translate the errors in some documented error codes.

#5 - 2012-03-26 17:32 - Adrian Föder

like the idea. Am I right that would mean you can do

```
1 /**
2  * @FLOW3\Validate(type="NotEmpty")
3  * @FLOW3\Validate(type="RegularExpression", options={ "regularExpression"="/^[mf]$/", "code"="UserGender" })
4  * @var string
5  * @ORM\Column(length=1)
6  */
7  protected $gender;
```

```
1<!-- ValidationErrors.de.xlf -->
```

```
2<trans-unit id="UserGender" xml:space="preserve" ....
```

?

#6 - 2012-03-27 11:13 - Christopher Hlubek

Yes, that's the idea. Right now it's not possible to treat one validator type differently regarding translations. Which makes the RegularExpressionValidator pretty useless for frontend development.

Adrian Föder wrote:

```
| like the idea. Am I right that would mean you can do
| [...]
|
```