TYPO3.Flow - Feature #28399

Validation message and code should be configurable for bundled validators

2011-07-22 17:36 - Christopher Hlubek

Status:	Needs Feedback	**Start date:**	2011-07-22
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Validation	**Estimated time:**	0.00 hour
Target version:			
PHP Version:		**Complexity:**	
Has patch:	No		

Description

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).

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. Additionaly a message type ("error" or "warning") would be nice.

The code change is pretty straight forward and should be compatible to the current version:

```
protected function isValid($value) {
  ...
  // If invalid
  $message = isset($this->options['message']) ? $this->options['message'] :
'Default error message';
  $code = isset($this->options['code']) ? $this->options['code'] : 12345678;
  $this->addError($message, $code);
}
```

Related issues:

Related to TYPO3.Flow - Feature #6456: Customization/localization of validato...	**Closed**	
Related to TYPO3.Flow - Feature #33970: Add translation files for bundled val...	**Resolved**	**2012-02-16**
Related to TYPO3.Flow - Feature #43841: Add package support to validation errors	**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

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

```
<!-- ValidationErrors.de.xlf -->
<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
> [...]