TYPO3.Flow - Feature # 3092

Status:	Resolved	**Priority:**	Must have
Author:	Robert Lemke	**Category:**	MVC
Created:	2009-04-15	**Assigned To:**	Robert Lemke
Updated:	2010-10-20	**Due date:**	

PHP Version:	
Has patch:	
Complexity:	
Subject:	Support annotation-based validation rules in controller actions

Description

It should be possible to define additional validation rules for arguments of a controller action by declaring them in an annotation:

```
  /**
   * The foo action
   *
   * @param string $firstArgument This argument contains an email address
   * @param integer $secondArgument This argument contains a number
   * @param F3\Foo\Bar\Model\Baz $thirdArgument This argument is automatically validated by the rules defined in the specified model
   * @return void
   * @validate $firstArgument EmailAddress
   * @validate $secondArgument NumberRange(minimum = 5, maximum = 100)
   */
  public function fooAction($firstArgument, $secondArgument, \F3\Foo\Bar\Model\Baz $thirdArgument) {
  }
```

Related issues:

related to TYPO3.Flow - Feature # 3153: Support of action based filter rules ...	**New**	**2009-04-22**

Associated revisions

Revision 9f137275 - 2009-04-16 08:22 - Robert Lemke

 - FLOW3: (MVC) Implemented the automatic validation rule registration based on @validate annotations in action methods. Note that validators are registered now but validation still does not happen (yet). Resolves #3092
 - FLOW3: (Validation) Modified the API of the ValidatorResolver

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.

History

#1 - 2009-04-16 09:00 - Robert Lemke

- Status changed from Accepted to Resolved

- % Done changed from 0 to 100

Applied in changeset r2159.