

TYPO3.Flow - Task #4268

handle model-attributes, that are not of type string, but user-input

2009-08-18 18:41 - Irene Höppner

Status:	Rejected	Start date:	
Priority:	Must have	Due date:	
Assignee:		% Done:	0%
Category:	MVC	Estimated time:	6.00 hours
Target version:		Has patch:	
Sprint:		Complexity:	
PHP Version:			

Description

I'm not sure, if this is a problem or something, I just didn't understand yet...

We have some attribute ("orderdate") in the model that is of type \DateTime and we want the user to add a date for that in the new-/edit-forms.

What we did and what happened:

1. Added the attribute + getter-/setter-method to the model class and an input-field to new.html:

```
/**
 *
 * @var \DateTime
 */
protected $orderdate = '';

/**
 *
 * @param \DateTime $orderdate
 * @return void
 */
public function setOrderdate(\DateTime $orderdate) {
    $this->orderdate = $orderdate;
}

/**
 *
 * @return \DateTime
 */
public function getOrderdate() {
    return $this->orderdate;
}

<label for="orderdate">Orderdate</label><br />
<f:form.textbox property="orderdate" id="orderdate" />
```

Result: Exception: Catchable Fatal Error: Argument 1 passed to F3\Blog\Domain\Model\Blog::setOrderdate() must be an instance of DateTime, string given in C:\Programme\xampp\htdocs\flow3\blog\Packages\Application\Blog\Classes\Domain\Model\Blog.php line 127

2. Removed the type hint from the setter:

```
/**
 *
 * @param \DateTime $orderdate
 * @return void
 */
public function setOrderdate($orderdate) {
```

```
$this->orderdate = $orderdate;
}
```

Result: Exception: 1244465558: Expected property of type DateTime, but got string

3. Added a validator the the attribute:

```
/**
 *
 * @var \DateTime
 * @validate DateTime
 */
protected $orderdate = '';
```

Result: error-message:

orderdate: The given subject was not a valid DateTime. Got: "string"

But: We get that message, no matter what we entered into the input-field.

4. Created the \DateTime-object in the setter, if the parameter is of type string:

```
/**
 *
 * @param \DateTime $orderdate
 * @return void
 */
public function setOrderdate($orderdate) {
    if (is_string($orderdate)) {
        $orderdate = new \DateTime($orderdate);
    }
    $this->orderdate = $orderdate;
}
```

Result: Cool, works - as long as the string in the input-form has a proper format like yyyy-mm-dd.

Otherwise another exception: DateTime::__construct() [datetime.--construct]: Failed to parse time string (fgsdasdfa) at position 0 (f): The timezone could not be found in the database

Ok, could check the string for a proper format in the setter... but we need the same check for several attributes in several classes...

5. What about writing our own validator?

Doesn't help because the setter comes first and needs a correct string already to create the datetime-object. Same exception like above.

6. Entered the orderdate-field to edit.html

```
<label for="orderdate">Orderdate</label><br />
<f:form.textbox property="orderdate" id="orderdate" />
```

Result: Exception: Warning: htmlspecialchars() expects parameter 1 to be string, object given in

C:\Programme\xampp\htdocs\flow3\blog\Packages\Framework\Fluid\Classes\Core\ViewHelper\TagBuilder.php line 161

Added some short hand Fluid-syntax helps in that case:

```
<label for="orderdate">Auftragsdatum:<br />
<f:form.textbox property="orderdate" value="{f:format.date('{job.orderdate}' format='d.m.Y')}" />
Format: dd.mm.yyyy<br />
```

Conclusion:

There has to be done some conversion between the strings coming from html-forms and attributes of the model that are not of type string. Before that conversion the strings probably have to be validated.

No idea where to do that best :-).

History

#1 - 2009-08-24 15:45 - Karsten Dambekalns

Well, the problem here is, that everything coming from a form is string, and PHP manages to convert to other primitive types if needed. For objects we have rules (if an UUID is found, we retrieve the matching object, ...) but we don't do anything for DateTime objects. Obviously we should.

Bottom line: you didn't do anything fundamentally wrong. There probably should be some nice Fluid helper to take a DateTime and produce a nice way to enter a date, combined with some magic on the server to transform input to DateTime.

#2 - 2010-01-20 17:09 - Robert Lemke

- Assignee set to Karsten Dambekalns

#3 - 2010-03-16 10:12 - Michael Sauter

I'm getting the same error for integers as well (with latest SVN - r3949). The setup is also the same, the number is entered in a TextFieldViewHelper.

The proposed "solution" (converting the variable in the setter method to integer) works, but without I'm getting Exception: 1244465559: Expected property of type integer, but got string

#4 - 2010-06-16 17:19 - Karsten Dambekalns

- Status changed from New to Accepted
- Priority changed from -- undefined -- to Must have
- Start date deleted (2009-08-18)
- Estimated time set to 6.00 h

#5 - 2010-08-18 15:16 - Karsten Dambekalns

- Category set to MVC

#6 - 2010-09-02 14:42 - Karsten Dambekalns

- Status changed from Accepted to New
- Assignee deleted (Karsten Dambekalns)

#7 - 2011-08-04 08:50 - Sebastian Kurfuerst

- Status changed from New to Rejected
- translation missing: en.field_remaining_hours set to 6.0

all the described stuff is possible with the new property mapper / type converters.