

## TYPO3.Flow - Task #45253

### Throw exception in PointcutMethodNameFilter if given method's argument does not match the actual method signature

2013-02-07 18:07 - Adrian Förder

<b>Status:</b>	Accepted	<b>Start date:</b>	2013-02-07
<b>Priority:</b>	Must have	<b>Due date:</b>	
<b>Assignee:</b>	Christian Müller	<b>% Done:</b>	0%
<b>Category:</b>	Security	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Has patch:</b>	No
<b>Sprint:</b>		<b>Complexity:</b>	
<b>PHP Version:</b>			

#### Description

Consider a TYPO3.Comment CommentRepository. I have set the following security resource:

```
[...]
'method(TYPO3\Comments\Domain\Repository\CommentRepository->remove(comment.author === current.securityContext.party))'
```

The CommentRepository itself does not carry an own remove() method, it inherits from TYPO3\Flow\Persistence\Repository as common.

In that mentioned case, NO proxy method is generated, hence no security applied.

When leaving the runtime constraint away,

```
[...]
'method(TYPO3\Comments\Domain\Repository\CommentRepository->remove())'
```

and still having not a concrete remove() method, but the inherited, it **works**, a proxy method is generated.

Third example: putting the runtime constraint back in,

```
[...]
'method(TYPO3\Comments\Domain\Repository\CommentRepository->remove(comment.author === current.securityContext.party))'
```

and putting a "dummy" remove() method into the concrete CommentRepository itself,

```
/**
 * @param \TYPO3\Comments\Domain\Model\Comment $comment
 */
public function remove($comment) {
    parent::remove($comment);
}
```

it again works, too!

If you want to test and fiddle, I prepared a repository at <https://github.com/afoeder/TYPO3.Comments>, just see the (only) Functional Test there and the Policy.yaml configuration. Maybe you want to checkout HEAD^1 there.

#### History

#1 - 2013-03-07 16:33 - Adrian Förder

- Tracker changed from Bug to Task

- Subject changed from Pointcuts on methods being inherited, carrying a runtime constraint, don't match to Throw exception in PointcutMethodNameFilter if given method's argument does not match the actual method signature

- Status changed from New to Closed

- Assignee set to Christian Müller

the wrongly, from my side, written Policy entry read ...->remove(comment.author === current.securityContext.party). This has been ignored in the PointcutMethodNameFilter->matches() method because the (inherited!) remove method did not expect a comment argument, but an "object" argument since the parent repository has this signature.

So the correct Policy entry would have been ->remove(object.author === ...)

However, an exception should be thrown in that case because the given argument does not match the actual method signature.

**#2 - 2013-03-07 16:33 - Adrian Förder**

- Status changed from Closed to Accepted