TYPO3 Core - Bug #61193

DependencyInjection with @var uses everything as classname

2014-08-26 12:17 - Guido Wehner

Status:	Rejected	**Start date:**	2014-08-26
Priority:	Won't have this time	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Extbase	**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	6.2	**Complexity:**	
PHP Version:		**Is Regression:**	No
Tags:		**Sprint Focus:**	

Description

Hello,

this is my first issue.

If you use dependency injection in your controller und you use @var to describe the type of the variable the ClassInfoFactory uses everything after @var as class name

In function buildClassInfoFromClassname($className) the className isn't parsed or formatted.

I need to write the name of the property after the type

Reference for phpdoc @var
[[http://phpdoc.org/docs/latest/references/phpdoc/tags/var.html]]

I need to write

```
/**
 * @var \BM\Simpleblog\Domain\Repository\BlogRepository $blogRepository
 * @inject
 */
protected $blogRepository;
```

but then "BM\Simpleblog\Domain\Repository\BlogRepository $blogRepository" is used as className.

$className = preg_replace("/(\s.*)/", "", $className);
would do the job, but I cannot say if it breaks tests or why there is no parsing the variable, maybe there are reasons I dont know.

History

#1 - 2014-08-28 15:20 - Alexander Opitz

- Category set to Extbase

#2 - 2014-09-05 19:50 - Frank Naegler

within in the class properties you can declare it like this:

```
/**
 * @var \BM\Simpleblog\Domain\Repository\BlogRepository
 * @inject
 */
protected $blogRepository;
```

simply remove the $blogRepository after the class name.
I agree this is a bug, but for the most IDEs and for TYPO3 the variable after the class is not needed.

#3 - 2014-09-05 20:01 - Frank Naegler

it looks like the problem is located in TYPO3\CMS\Extbase\Reflection\DocCommentParser::parseTag($line) the tag is split with a limit: $tagAndValue

= preg_split('/\\s/', $line, 2);.
I am not sure why the limit is set, maybe someone from the extbase team can get an answer.

#4 - 2014-09-06 12:39 - Helmut Hummel

- Status changed from New to Rejected

- Priority changed from Should have to Won't have this time

@var \BM\Simpleblog\Domain\Repository\BlogRepository $blogRepository

This is an invalid property annotation. I do not think we need to support invalid annotations.

#5 - 2014-09-06 12:41 - Frank Naegler

Why it is an invalid annotation? The documentation for @var is

```
@var [“Type”] [$element_name] [<description>]
```

see the link above in the original issue message.

#6 - 2015-02-09 13:23 - Nils Blattner

http://www.phpdoc.org/docs/latest/references/phpdoc/tags/var.html

Clearly shows that the above is a valid notation and should be interpreted correctly.
Can you please reopen the ticket?

Kind regards
Nils

#7 - 2015-02-10 01:35 - Christian Kuhn

- Status changed from Rejected to New

#8 - 2016-03-06 02:19 - Mathias Schreiber

- Status changed from New to Rejected

Taken from the mentioned docs:

```
class Foo
{
  /** @var string|null Should contain a description */
  protected $description = null;
}
```

We support this.

Your misconception ($varname goes in as well):

```
class Foo
{
  /**
   * @var string $name        Should contain a description
   * @var string $description Should contain a description
   */
  protected $name, $description;
}
```

Used for **compound** syntax (which we do not support).
Helmut wins :)