

TYPO3.Flow - Feature #31788

[PERFORMANCE] cache implementation for AnnotationReader in Flow3AnnotationDriver

2011-11-13 16:54 - rottenrice no-lastname-given

Status:	Rejected	Start date:	2011-11-13
Priority:	Should have	Due date:	
Assignee:	Karsten Dambekalns	% Done:	0%
Category:	Reflection	Estimated time:	0.00 hour
Target version:		Complexity:	
PHP Version:			
Has patch:	No		

Description

doctrine has classes to cache the parsed annotations. with cached annotations you can boost the parsetime by ~200ms (=15%, tested with TYPO3.Blog).

some caching strategies:

```
$this->reader = new \Doctrine\Common\Annotations\CachedReader(  
    new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationRead  
er()),  
    new \Doctrine\Common\Cache\ApcCache()  
);
```

```
$this->reader = new \Doctrine\Common\Annotations\CachedReader(  
    new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationRead  
er()),  
    new \Doctrine\Common\Cache\ArrayCache()  
);
```

```
$this->reader = new \Doctrine\Common\Annotations\FileCachedReader(  
    new \Doctrine\Common\Annotations\IndexedReader(new \Doctrine\Common\Annotations\AnnotationRead  
er()),  
    CACHING_DIR  
);
```

it would be better to use the Objects.yaml to set the caching strategy instead of using __construct()

```
/**  
 * @var \Doctrine\Common\Annotations\AnnotationReader  
 */  
protected $reader;
```

i think it is better to use the interface \Doctrine\Common\Annotations\Reader

```
--  
class: TYPO3\FLOW3\Persistence\Doctrine\Mapping\Driver\Flow3AnnotationDriver
```

History

#1 - 2011-11-25 11:32 - Christopher Hlubek

+1 for the interface. We should implement a custom CachedReader using the FLOW3 caching framework. Should be fairly easy.

#2 - 2011-12-14 15:40 - Karsten Dambekalns

- Category set to Reflection

- Status changed from New to Needs Feedback

- Assignee set to Karsten Dambekalns

Christopher Hlubek wrote:

+1 for the interface. We should implement a custom CachedReader using the FLOW3 caching framework. Should be fairly easy.

The strategy currently is to cache the results the ReflectionService builds with the reader. No doubled caching, if it can be avoided.

#3 - 2012-03-07 16:07 - Karsten Dambekalns

- Status changed from Needs Feedback to Rejected

Recent improvements have shown the caching we have in place does the job.