

TYPO3.Flow - Bug #45407

charset option for mysql connection is ignored with PHP < 5.3.6

2013-02-12 12:36 - Benjamin Albrecht

Status:	Closed	Start date:	2013-02-12
Priority:	Must have	Due date:	
Assignee:	Karsten Dambekalns	% Done:	0%
Category:	Persistence	Estimated time:	0.00 hour
Target version:		Complexity:	
PHP Version:	5.3		
Has patch:	No		

Description

Hi,

the configurable charset option for the backed (default "utf8") is ignored with PHP Version < 5.3.6
But the systems requirements of Flow 2.0 are PHP 5.3.2
(Here is the link to the PHP documentation: <http://php.net/manual/de/ref.pdo-mysql.connection.php>)

There are several implications with this Bug:

- Upgrade of PHP Version to 5.3.6 is not possible

We are using Debian 6.0 with the latest PHP 5.3 Debian package. This is PHP 5.3.3-7+squeeze14, wich contains 5.3.3 plus Bugfixes and Securityfixes, but not new features, like the pdo_mysql charset support.

- If we ignore this bug and writing content into the db only with Flow, there are no wrong character at the frontend / website. But the db contains broken characters. After an update to an new php version or another server, the broken characters are shown at the frontend... (same after upgrading from FLOW3 1.1)

The mysql driver of doctrine supports "driverOptions" to set something like "SET NAMES utf8", but after this commit and using the native doctrine sources, this is no longer useful...
<http://git.typo3.org/FLOW3/Packages/Doctrine.DBAL.git/commitdiff/b231de400af149a7e6a980939b067c6177707e3d>

Another useful solution could be the "SessionInitialization" of doctrine, but after this commit a great configure option of Flow is removed :-/ <http://git.typo3.org/FLOW3/Packages/TYPO3.FLOW3.git/commit/0636682f3b304c6a63e7b1558d7d32c153c1940f>
Why?

History

#1 - 2013-02-16 20:39 - Ottmar Biebersdorf

I can confirm this issue for Debian Squeeze with PHP 5.3.3-7+squeeze14. This is a serious issue as the data in the DB-Dump is corrupt (no DB-Backup possible).

Like Benjamin Albrecht suggested, using Doctrine SessionInitialization with 'SET NAMES utf8' seems to fix this issue.

#2 - 2013-02-17 15:05 - Benjamin Albrecht

Hi,

we are using an AOP aspect to set the doctrine SQLSessionInit as a workaround:
With this solution it's not necessary to patch the core and the data in your DB is valid ;-)

```
<?php
namespace Your\Package\Aspect;

use TYPO3\Flow\Annotations as Flow;

/**
 * @Flow\Aspect
 */
class EntityManagerFactoryAspect {
    /**
     * @Flow\Around("method(TYPO3\Flow\Persistence\Doctrine\EntityManagerFactory->create())")
     * @param \TYPO3\Flow\Aop\JoinPointInterface $joinPoint The current join point
     * @return \Doctrine\ORM\EntityManager
     */
    public function sqlSessionInitAdvice(\TYPO3\Flow\Aop\JoinPointInterface $joinPoint) {
        // @var $entityManager \Doctrine\ORM\EntityManager
    }
}
```

```

$entityManager = $joinPoint->getAdviceChain()->proceed($joinPoint);
// @var $eventManager \Doctrine\Common\EventManager
$eventManager = $entityManager->getEventManager();

$sql = "SET NAMES utf8;";

$sqlSessionInit = new \Doctrine\DBAL\Event\Listeners\SQLSessionInit($sql);

$eventManager->addEventSubscriber($sqlSessionInit);

return $entityManager;
}
}
?>

```

This github gist gave us the inspiration to our solution: <https://gist.github.com/bwaidelich/3917639>
(thx to @khuppenbauer on twitter for this hint)

#3 - 2013-08-07 14:20 - Daniel Ostmann

You can use this configuration in your settings file:

```

TYPO3:
  Flow:
    persistence:
      backendOptions:
        driverOptions:
          1002: 'SET NAMES utf8'

```

Since we can't use constants in the configuration file we use it's value of 1002 (PDO::MYSQL_ATTR_INIT_COMMAND).

See also: <http://www.php.net/manual/de/pdo.construct.php#96325>

#4 - 2013-08-14 16:23 - Karsten Dambekalns

- Status changed from New to Closed
- Assignee set to Karsten Dambekalns

This is an issue we had fixed in our own version of Doctrine 2 DBAL, see <https://review.typo3.org/#/c/7746/>

But since we use the unmodified version now, this is an issue that needs to be fixed with Doctrine, as notes in <http://www.doctrine-project.org/jira/browse/DBAL-351> which points to <https://github.com/doctrine/dbal/pull/206>. This one was dismissed though, pointing out a solution.

The use of driverOptions is probably the easiest way (short of upgrading PHP).

#5 - 2013-09-26 10:49 - Kay Strobach

This Option should be the default for MySQL ... overriding is ok, but this is definitely a problem on many hosts!
Atleast it should be in the example yaml file ...