

TYPO3.Flow - Bug # 42955

Status:	Resolved	Priority:	Should have
Author:	René Pflamm	Category:	Persistence
Created:	2012-11-14	Assigned To:	Bastian Waidelich
Updated:	2012-12-11	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: FLOW3 1.1.0			
Subject:	QueryResult::count() returns 0 for queries with offset		
Description			
NOTE See comments below			
On my Project I notice if I have more then itemsPerPage and switched to another Page this one ist Empty.			
As example:			
On first page, the object to paginate has an count of 27. The paginatedObject hast a count of 27 but display only 10.			
With an itemsPerPage limit of 10 I got 3 pages. On Page 2 and 3 I got an paginatedObject of zero and display nothing.			
Currently I hotfixed this quick and dirty with an replace of \$modifiedObjects = \$query->execute(); to \$modifiedObjects = \$query->getResult(); (see attachment).			
Related issues:			
duplicates TYPO3.Flow - Bug # 41146: Query with setOffset returns count() = 0		Resolved	2012-09-21

Associated revisions

Revision d46eef43 - 2012-11-27 16:38 - Bastian Waidelich

[!!!][BUGFIX] Respect limit/offset constraints when counting results

Until now QueryResult::count() discarded limit constraints that were active for the affected query, so

`$query->setLimit(3)->execute()->count()` could return a value > 3.

If the query had an offset Flow even threw an exception.

This changeset adjusts the Query so that limit and offset constraints are taken into account. It furthermore unsets the offset in the query in order to circumvent the exception.

This is a breaking change if you relied on Query::count() to always return **all** results regardless of a specified limit. In this case better generate multiple QueryResults (they're only created in-memory until you actually access them)

Change-Id: I7ed06adf54d7ce9d5aff08433afc4aee9964e8b7

Fixes: #42955

Releases: 1.1, 1.2

Revision d7bf9fca - 2012-12-10 21:42 - Bastian Waidelich

[!!!][BUGFIX] Respect limit/offset constraints when counting results

Until now QueryResult::count() discarded limit constraints that were active for the affected query, so
\$query->setLimit(3)->execute()->count() could return a value > 3.
If the query had an offset Flow even threw an exception.

This changeset adjusts the Query so that limit and offset constraints are taken into account. It furthermore unsets the offset in the query in order to circumvent the exception.

This is a breaking change if you relied on Query::count() to always return **all** results regardless of a specified limit. In this case better generate multiple QueryResults (they're only created in-memory until you actually access them)

Change-Id: I7ed06adf54d7ce9d5aff08433afc4aee9964e8b7

Fixes: #42955

Releases: 1.1, 1.2

History

#1 - 2012-11-14 11:37 - Bastian Waidelich

- Category set to Widgets
- Status changed from New to Needs Feedback
- Assigned To set to Bastian Waidelich

Hi René,

| On my Project I notice if I have more then itemsPerPage and switched to another Page this one ist Empty.

I don't completely understand. Could you post a snippet of the affected template, controller & repository code?
The method getResult() is not defined in the QueryInterface so it can't be used here.

Best,
Bastian

#2 - 2012-11-14 11:58 - René Pflamm

The controller:

```
public function indexAction() {  
  
    $this->view->assignMultiple(array(  
        'companys' => $this->companyRepository->findByFilter($this->companyFilter),  
        'companyFilter' => $this->companyFilter  
    )  
    );  
};
```

```
}
```

(\$this->companyRepository->findByAll() is also effected, companyFilter is an SessionObject)

The findByFilter function:

```
/**
 * @param \My\Package\Domain\Model\CompanyFilter $companyFilter
 * @return \TYPO3\FLOW3\Persistence\QueryResultInterface
 */
public function findByFilter($companyFilter = NULL) {
    $query = $this->createQuery();
    $constraints = array();

    if ($companyFilter->getSearchWord()) {
        $constraints[] = $query->logicalOr(
            /* Here are some Likes */
        );
    }

    if ($companyFilter->getMyProperty()) {
        $constraints[] = $query->equals('myProperty.title', $companyFilter->getMyProperty());
    }

    if (count($constraints) > 0) $query->matching($query->logicalAnd($constraints));

    return $query->execute();
}
```

The template:

```
<f:widget.paginate objects="{companys}" as="paginatedCompanys" configuration="{itemsPerPage: 10, insertAbove: 0, insertBelow: 1,
maximumNumberOfLinks: 10}" widgetId="companyListPaginate">
<table id="companyList">
...
<f:if condition="{paginatedCompanys}">
<f:then>
<f:for each="{paginatedCompanys}" as="company" iteration="companyIterator">
</f:then>
<f:else>
nothing to show
</f:else>
</f:if>
...
</f:for>
...
</table>
```

```
</f:widget.paginate>
```

On Page one it shows 10 items for `<f:for each="{paginatedCompanies}">`

On Page two and three I got the else function with "nothing to show".

With the count ViewHelper I got:

```
<f:widget.paginate objects="{companies}" as="paginatedCompanies" configuration="{itemsPerPage: 10, insertAbove: 0, insertBelow: 1,
maximumNumberOfLinks: 10}" widgetId="companyListPaginate">
  {companies -> f:count()} => {paginatedCompanies ->f:count}
</f:widget.paginate>
```

Page one:

```
27 => 27
```

Page two:

```
27 => 0
```

Page three:

```
27 => 0
```

#3 - 2012-11-14 12:08 - René Pflamm

- File 20121114120400b95313.txt added

I found out, an Exception is logged:

Uncaught exception in line 106 of /myPath/Packages/Framework/Doctrine\ORM/Classes/Internal/Hydration/AbstractHydrator.php: No result was found for query although at least one row was expected.

#4 - 2012-11-14 12:08 - Bastian Waidelich

Hi,

thanks for the details.

I need to test this, but could you try the following for now:

```
1 <f:if condition="{companies}">
2 <f:then>
```

2015-08-04

4/9

```

3 <f:widget.paginate objects="{companys}" as="paginatedCompanys" configuration="{itemsPerPage: 10, insertAbove: 0, insertBelow: 1,
maximumNumberOfLinks: 10}" widgetId="companyListPaginate">
4 <table id="companyList">
5 ...
6 <f:for each="{paginatedCompanys}" as="company" iteration="companyIterator">
7 ...
8 </f:for>
9 </table>
10 </f:widget.paginate>
11 </f:then>
12 <f:else>
13 Nothing to show
14 </f:else>
15</f:if>

```

if {companys} is set, {paginatedCompanys} will always be set, too.

btw: it should be "companIEs" ;)

#5 - 2012-11-14 12:25 - René Pflamm

Okay, this variant seems to be working... but why? ^^

now I need to rewrite the paginate template ;)

thanks for you note about companIEs ;)

#6 - 2012-11-21 16:45 - Bastian Waidelich

- Subject changed from *PaginateController doesn't execute query correct* to *QueryResult::count() returns 0 for queries with offset*

#7 - 2012-11-21 16:50 - Bastian Waidelich

- Project changed from *TYPO3.Fluid* to *TYPO3.Flow*

- Category deleted (*Widgets*)

- Assigned To deleted (*Bastian Waidelich*)

Ok, the actual issue with this is that the Doctrine/QueryResult::count() throws an exception if the underlying query has an offset (which is the case for queries that have been modified through the pagination widget).

Internally the error happens in Doctrine\Query::count():

```

1 public function count() {
2     try {
3         $originalQuery = $this->queryBuilder->getQuery();
4         $dqlQuery = clone $originalQuery;
5         $dqlQuery->setParameters($originalQuery->getParameters());
6         $dqlQuery->setHint(\Doctrine\ORM\Query::HINT_CUSTOM_TREE_WALKERS, array('TYPO3Flow\Persistence\Doctrine\CountWalker'
));
7         return (int)$dqlQuery->getSingleScalarResult();

```

```

8 } catch (\Doctrine\ORM\ORMException $ormException) {
9     $this->systemLogger->logException($ormException);
10    return 0;
11 } catch (\PDOException $pdoException) {
12     throw new DatabaseConnectionException($pdoException->getMessage(), $pdoException->getCode());
13 }
14}

```

Where the exception is caught and 0 is returned. If I reset the offset here with

```
$dqlQuery->setFirstResult(NULL);
```

it seems to work.

@Karsten can you verify this?

#8 - 2012-11-21 16:50 - Bastian Waidelich

Ok, the actual issue with this is that the Doctrine/QueryResult::count() throws an exception if the underlying query has an offset (which is the case for queries that have been modified through the pagination widget).

Internally the error happens in Doctrine\Query::count():

```

1 public function count() {
2     try {
3         $originalQuery = $this->queryBuilder->getQuery();
4         $dqlQuery = clone $originalQuery;
5         $dqlQuery->setParameters($originalQuery->getParameters());
6         $dqlQuery->setHint(\Doctrine\ORM\Query::HINT_CUSTOM_TREE_WALKERS, array('TYPO3\Flow\Persistence\Doctrine\CountWalker'
7     ));
8         return (int)$dqlQuery->getSingleScalarResult();
9     } catch (\Doctrine\ORM\ORMException $ormException) {
10        $this->systemLogger->logException($ormException);
11        return 0;
12    } catch (\PDOException $pdoException) {
13        throw new DatabaseConnectionException($pdoException->getMessage(), $pdoException->getCode());
14    }
15}

```

Where the exception is caught and 0 is returned. If I reset the offset here with

```
$dqlQuery->setFirstResult(NULL);
```

it seems to work (even though counting on a query with limit will always count all results..)

@Karsten can you verify this?

#9 - 2012-11-21 16:53 - Bastian Waidelich

- Category set to Persistence
- Assigned To set to Bastian Waidelich
- Target version set to 2.0 beta 1

#10 - 2012-11-21 16:57 - Karsten Dambekalns

Bastian Waidelich wrote:

| @Karsten can you verify this?

Didn't try it, but it sounds reasonable. Whatever happens inside Doctrine there - resetting the offset for this case should not have any side effects.

#11 - 2012-11-21 17:18 - Bastian Waidelich

Karsten Dambekalns wrote:

| Didn't try it, but it sounds reasonable. Whatever happens inside Doctrine there
| resetting the offset for this case should not have any side effects.

Thanks.

I could push a change set, but I just wonder if that behavior is correct at all:

```
$query->setLimit(10);  
$query->setOffset(5);  
$query->execute()->count(); // will throw away limit & offset
```

Probably the right way to do this would be to really count the results as soon as an offset is set and compare return $\max(\$numberOfRows, \$limit)$...

#12 - 2012-11-21 17:56 - Bastian Waidelich

Bastian Waidelich wrote:

| Probably the right way to do this would be to really count the results as soon as an offset
| is set and compare return $\max(\$numberOfRows, \$limit)$...

This (rough) adjustment does this:

```
1 public function count() {  
2   try {  
3     $originalQuery = $this->queryBuilder->getQuery();
```

```

4   $dqlQuery = clone $originalQuery;
5   $dqlQuery->setParameters($originalQuery->getParameters());
6   $dqlQuery->setHint(\Doctrine\ORM\Query::HINT_CUSTOM_TREE_WALKERS, array('TYPO3\Flow\Persistence\Doctrine\CountWalker'
));
7   $offset = $dqlQuery->getFirstResult();
8   $limit = $dqlQuery->getMaxResults();
9   if ($offset !== NULL) {
10    $dqlQuery->setFirstResult(NULL);
11  }
12  $numberOfResults = (int)$dqlQuery->getSingleScalarResult();
13  if ($offset !== NULL) {
14    $numberOfResults = max(0, $numberOfResults - $offset);
15  }
16  if ($limit !== NULL) {
17    $numberOfResults = min($numberOfResults, $limit);
18  }
19  return $numberOfResults;
20 } catch (\Doctrine\ORM\ORMException $ormException) {
21   $this->systemLogger->logException($ormException);
22   return 0;
23 } catch (\PDOException $pdoException) {
24   throw new DatabaseConnectionException($pdoException->getMessage(), $pdoException->getCode());
25 }
26}

```

#13 - 2012-11-27 15:56 - Bastian Waidelich

- Status changed from Needs Feedback to Accepted

#14 - 2012-11-27 16:38 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <http://review.typo3.org/16814>

#15 - 2012-11-29 12:35 - Bastian Waidelich

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:d46eef43f377c15bd8987bc1c61d519e868d875a.

#16 - 2012-12-10 21:37 - Gerrit Code Review

- Status changed from Resolved to Under Review

Patch set 1 for branch **FLOW3-1.1** has been pushed to the review server.

It is available at <https://review.typo3.org/17074>

#17 - 2012-12-10 21:42 - Gerrit Code Review

Patch set 2 for branch **FLOW3-1.1** has been pushed to the review server.

It is available at <https://review.typo3.org/17074>

#18 - 2012-12-11 10:27 - Karsten Dambekalns

- Status changed from Under Review to Resolved

Files

PaginateController.patch	827 Bytes	2012-11-14	René Pflamm
20121114120400b95313.txt	5.9 kB	2012-11-14	René Pflamm