

Database - Bug #11931

NPE in BatchResults when listing no rows

10/02/2026 05:30 AM - Alexandru Lungu

Status:	Test	Start date:	
Priority:	Normal	Due date:	
Assignee:	Teodor Gorghe	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	Alexandru Lungu
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#1 - 10/02/2026 05:33 AM - Alexandru Lungu

- Status changed from New to WIP
- Assignee set to Teodor Gorghe

I encountered a NPE as:

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.persist.BatchResults.fetchBatchByIds (BatchResults.java:540)
    at com.goldencode.p2j.persist.BatchResults.fetchBatch (BatchResults.java:471)
    at com.goldencode.p2j.persist.BatchResults.first (BatchResults.java:329)
    at com.goldencode.p2j.persist.PreselectQuery.first (PreselectQuery.java:3393)
    at com.goldencode.p2j.persist.AbstractQuery.wrapFirst (AbstractQuery.java:4782)
    at com.goldencode.p2j.persist.AbstractQuery.getFirst (AbstractQuery.java:2938)
    at com.goldencode.p2j.persist.QueryWrapper.getFirst (QueryWrapper.java:6344)
```

Mind that Persistence.list yields null is the result-set is empty, thus Map<Long, Object[]> byPk = new HashMap<>(fetched.size() * 2); throws a NPE. Also for (Object elem : fetched) might be problematic. There are multiple uses of list, but all are null guarded. (e.g. AdaptiveQuery.repositionByID, SimpleResults.<init>, RB.aggregate, etc.). BatchResults seems to be the only example where this doesn't happen.

#3 - 10/02/2026 05:36 AM - Teodor Gorghe

Ok, I understand the issue.

Writing the testcases and the fix.

#4 - 10/02/2026 08:40 AM - Teodor Gorghe

Reproduced the issue.

Create task branch **11931a** and committed revision **16781**:

- Fixed NPE when all records of a batch were deleted (Persistence.list returns null).

Alexandru, please review.

#5 - 10/02/2026 08:40 AM - Teodor Gorghe

- Status changed from WIP to Review

- % Done changed from 0 to 100

- reviewer Alexandru Lungu added

#6 - 10/07/2026 07:58 AM - Alexandru Lungu

```
+         if (fetched == null)
+         {
+             // every record of this component in the batch was deleted after caching
+             fetched = Collections.emptyList();
+         }
```

Isn't this approach going to create a redundant list in memory, so that `Map<Long, Record> entityMap = new HashMap<>(fetched.size() * 2);` creates an empty map in memory and for (Object elem : fetched) is no-op?

I would prefer to short-circuit the upcoming code completely if the fetched is null, so that we don't do redundant allocations.

#7 - 10/07/2026 08:01 AM - Teodor Gorghe

I have thought that this is a reference for an immutable object, and therefore, no redundant memory allocation is being done.

I will do the short-circuit approach.

#8 - 10/07/2026 08:05 AM - Alexandru Lungu

Teodor Gorghe wrote:

I have thought that this is a reference for an immutable object, and therefore, no redundant memory allocation is being done.

Ah good point. Then lets have it this way.

#9 - 10/07/2026 08:05 AM - Alexandru Lungu

- *Status changed from Review to Internal Test*

I am fine with the approach, please continue with some light smoke tests. When ready, please let me know to approve the merge.

#10 - 10/07/2026 08:16 AM - Teodor Gorghe

I have already smoke-tested on an app and it has passed. Added more BatchResults tests in xfer testcases and all passes now.

#11 - 10/07/2026 08:35 AM - Alexandru Lungu

- *Status changed from Internal Test to Merge Pending*

Please merge 11931a to trunk now. Let me know what custom branches have BatchResults and may need these changes.

#12 - 10/07/2026 08:42 AM - Teodor Gorghe

- *Status changed from Merge Pending to Test*

Branch **11931a** was merged into trunk as rev. **16790** and archived.

Branches which uses BatchResults: **9866c**, **9866d**, **10614_main**, **11748a**.

#13 - 10/07/2026 08:48 AM - Alexandru Lungu

Teodor Gorghe wrote:

Branches which uses BatchResults: **9866c**, **9866d**

Radu, please assess.

10614_main

I will assess this.

11748a.

I will assess this.

#15 - 10/07/2026 09:08 AM - Radu Apetrii

Teodor Gorghe wrote:

Branches which uses BatchResults: **9866c**, **9866d**

9866d is not in use anymore, so there's nothing to worry about. 9866c is slowly reaching its end, and the customer asked for no more revisions ported to it. However, I will ask about it, since this is a relevant fix.

I would also like to mention **9866e**, that's the most influential right now. Is this something for me to check if BatchResults is used, or can Teodor do a quick check, since he knows what to look for?

#16 - 10/07/2026 09:12 AM - Teodor Gorghe

Sorry, I wanted to say **9866e**.

BatchResults is actually been used in that project, in fact, the original implementation is #9393 (which fixes #9972). That path doesn't drop records while iterating that table, and that's why no one has talked about it.