

xwiki / xwiki-platform Public

<> Code Pull requests 214 Actions Projects Security and quality 229

# Commit 47b568c

michitux committed on Dec 10, 2025 · ✓ 1 / 1

XWIKI-23550: DBListClass should respect query limit

- \* Respect the query limit in DBListClass#getDBList.
- \* Add a unit test.

master · xwiki-platform-18.2.1 ··· xwiki-platform-18.0.0-rc-1

1 parent 7d7a391 commit 47b568c

2 files changed +70 -1 lines changed

↑ Top ⚙️

Filter files...

- main/java/com/xpn/xwiki/objects/classes
      - DBListClass.java
    - test/java/com/xpn/xwiki/objects/classes
      - DBListClassTest.java

2 files changed +70 -1 lines changed

Search within code ⚙️

.../xpn/xwiki/objects/classes/DBListClass.java

```

@@ -32,7 +32,9 @@
32 32 import org.slf4j.Logger;
33 33 import org.slf4j.LoggerFactory;
34 34 import org.xwiki.component.util.DefaultParameterizedType;
35 + import org.xwiki.query.Query;
35 36 import org.xwiki.query.QueryBuilder;
37 + import org.xwiki.security.SecurityConfiguration;
36 38 import org.xwiki.security.authorization.AuthorExecutor;

```

```

37 39
38 40 import com.xpn.xwiki.XWiki;
@@ -118,13 +120,20 @@ public List<ListItem> getDBList(XWikiContext context)
118 120 List<ListItem> list = getCachedDBList(context);
119 121 if (list == null) {
120 122     try {
123 +         SecurityConfiguration securityConfiguration =
        Utils.getComponent(SecurityConfiguration.class);
121 124         DefaultParameterizedType dbListQueryBuilderType =
122 125             new DefaultParameterizedType(null, QueryBuilder.class,
        DBListClass.class);
123 126         QueryBuilder<DBListClass> dbListQueryBuilder =
        Utils.getComponent(dbListQueryBuilderType);
124 127         // Execute the query with the rights of the class last author.
125 128         AuthorExecutor authorExecutor =
        Utils.getComponent(AuthorExecutor.class);
126 129         list = makeList(authorExecutor.call(() -> {
127 -         return dbListQueryBuilder.build(this).execute();
130 +         Query query = dbListQueryBuilder.build(this);
131 +         int configuredLimit =
        securityConfiguration.getQueryItemsLimit();
132 +         // Limit unlimited queries or queries with a high limit to
        the configured limit.
133 +         if (configuredLimit > 0 && (query.getLimit() <= 0 ||
        query.getLimit() > configuredLimit)) {
134 +             query.setLimit(configuredLimit);
135 +         }
136 +         return query.execute();
128 137     }, getOwnerDocument().getAuthorReference(),
        getDocumentReference());
129 138     } catch (Exception e) {
130 139         LOGGER.warn("Failed to get the Database List values. Root cause
        is [{}].",

```

```

.../xwiki/objects/classes/DBListClassTest.java
@@ -21,9 +21,18 @@
21 21
22 22 import java.util.ArrayList;

```

```

23 23      import java.util.List;
24 24  + import java.util.concurrent.Callable;
24 25
25 26      import org.junit.jupiter.api.BeforeEach;
26 27      import org.junit.jupiter.api.Test;
28 28  + import org.junit.jupiter.params.ParameterizedTest;
29 29  + import org.junit.jupiter.params.provider.CsvSource;
30 30  + import org.xwiki.model.reference.DocumentReference;
31 31  + import org.xwiki.query.Query;
32 32  + import org.xwiki.query.QueryBuilder;
33 33  + import org.xwiki.security.SecurityConfiguration;
34 34  + import org.xwiki.security.authorization.AuthorExecutor;
35 35  + import org.xwiki.test.junit5.mockito.MockComponent;
27 36
28 37      import com.xpn.xwiki.XWikiContext;
29 38      import com.xpn.xwiki.doc.XWikiDocument;
30 39  @@ -34,7 +43,12 @@
34 43
35 44      import static org.junit.jupiter.api.Assertions.assertEquals;
36 45      import static org.mockito.ArgumentMatchers.any;
37 46  + import static org.mockito.ArgumentMatchers.anyInt;
37 47      import static org.mockito.Mockito.doAnswer;
38 48  + import static org.mockito.Mockito.mock;
39 49  + import static org.mockito.Mockito.never;
40 50  + import static org.mockito.Mockito.verify;
41 51  + import static org.mockito.Mockito.when;
38 52
39 53      /**
40 54      * Unit tests for {@link DBListClass}.
41 55  @@ -48,6 +62,15 @@ class DBListClassTest
48 62      @InjectMockitoOldcore
49 63      private MockitoOldcore oldcore;
50 64
51 65  + @MockComponent
52 66  + private QueryBuilder<DBListClass> queryBuilder;
53 67  +
54 68  + @MockComponent
55 69  + private SecurityConfiguration securityConfiguration;
56 70  +
57 71  + @MockComponent

```

```

72 + private AuthorExecutor authorExecutor;
73 +
51 74 @BeforeEach
52 75 void before()
53 76 {
    ⋮
    ⋮
    ⋮
@@ -399,4 +422,41 @@ void testReturnColWithInvalidQuery()
399 422 assertEquals("-", dblc.returnCol("do something", true));
400 423 assertEquals("-", dblc.returnCol("do something", false));
401 424 }
425 +
426 + @ParameterizedTest
427 + @CsvSource({
428 +     "10, 20, false",
429 +     "20, 10, true",
430 +     "0, 0, false",
431 +     "0, -1, false",
432 +     "20, -1, false",
433 +     "0, 20, true"
434 + })
435 + void getDBListLimit(int queryLimit, int configuredLimit, boolean setLimit)
    throws Exception
436 + {
437 +     Query mockQuery = mock();
438 +     when(mockQuery.getLimit()).thenReturn(queryLimit);
439 +     when(mockQuery.execute()).thenReturn(List.of());
440 +
    when(this.securityConfiguration.getQueryItemsLimit()).thenReturn(configuredLimit);
441 +
442 +     when(this.authorExecutor.call(any(), any(), any())).then(invocation ->
    {
443 +         Callable<?> callable = invocation.getArgument(0);
444 +         return callable.call();
445 +     });
446 +
447 +     XWikiDocument ownerDocument = new XWikiDocument(new
    DocumentReference("wiki", "space", "page"));
448 +     DBListClass dbListClass = new DBListClass();
449 +     dbListClass.setOwnerDocument(ownerDocument);

```

```
450 +
451 +     when(this.queryBuilder.build(dbListClass)).thenReturn(mockQuery);
452 +
453 +     dbListClass.getDBList(this.oldcore.getXWikiContext());
454 +
455 +     if (setLimit) {
456 +         verify(mockQuery).setLimit(configuredLimit);
457 +     } else {
458 +         verify(mockQuery, never()).setLimit(anyInt());
459 +     }
460 +     verify(mockQuery).execute();
461 + }
402 462 }
```

## Comments 0



Please [sign in](#) to comment.