dasgarner / **xibo-cms** Public

forked from xibosignage/xibo-cms

Code | Pull requests | Actions | Projects | Security and quality | Insights

Commit `b8d25fe`

dasgarner committed on Mar 5

```
DataSet: improve sanitization

 relates to xibosignageltd/xibo-private#1244
```

`hotfix/tempel`

1 parent 7d375b2 commit b8d25fe

3 files changed +122 -7

Top

Filter files...

- lib
  - Controller
    - DataSetData.php
  - Entity
    - DataSet.php
  - Helper
    - Sql.php

Search within code

lib/Controller/DataSetData.php

```
@@ -125,7 +125,7 @@ public function grid($dataSetId)
125  125          $filter = trim($filter, 'AND');
126  126
127  127          // Work out the limits
```

```
128      -          $filter = $this->gridRenderFilter(['filter' => $this->getSanitizer()->getParam('filter', $filter)]);
     128 +          $filter = $this->gridRenderFilter(['filter' => $filter]);
129  129
130  130            $this->getState()->template = 'grid';
131  131            $this->getState()->setData($dataSet->getData([
```

lib/Entity/DataSet.php

```
@@ -19,6 +19,7 @@
19   19   use Xibo\Factory\DataSetFactory;
20   20   use Xibo\Factory\DisplayFactory;
21   21   use Xibo\Factory\PermissionFactory;
     22 + use Xibo\Helper\Sql;
22   23   use Xibo\Service\ConfigServiceInterface;
23   24   use Xibo\Service\DateServiceInterface;
24   25   use Xibo\Service\LogServiceInterface;
@@ -189,9 +190,6 @@ class DataSet implements \JsonSerializable
189  190
190  191        private $countLast = 0;
191  192
192      -     /** @var array Blacklist for SQL */
193      -     private $blackList = array(';', 'INSERT', 'UPDATE', 'SELECT', 'DELETE', 'TRUNCATE', 'TABLE', 'FROM', 'WHERE');
194      -
195  193        /** @var  SanitizerServiceInterface */
196  194        private $sanitizer;
197  195
@@ -337,7 +335,7 @@ public function getUniqueColumnValues($columns)
337  335                  if ($column->heading == $heading) {
338  336                      // Formula column?
339  337                      if ($column->dataSetColumnTypeId == 2) {
340      -                         $select .= str_replace($this->blackList, '', htmlspecialchars_decode($column->formula, ENT_QUOTES)) . ' AS `' . $column->heading . '`,';
     338 +                         $select .= Sql::cleanup(htmlspecialchars_decode($column->formula, ENT_QUOTES)) . ' AS `' . $column->heading . '`,';
```

```
341 339                         }
342 340                         else {
343 341                             $select .= '`' . $column->heading . '`,';
    @@ -420,7 +418,7 @@ public function getData($filterBy = [], $options = [],
    $extraParams = [])
420 418                         continue;
421 419                     }
422 420
423     -                   $formula = str_replace($this->blackList, '',
    htmlspecialchars_decode($column->formula, ENT_QUOTES));
    421 +                   $formula = Sql::cleanup(htmlspecialchars_decode($column-
    >formula, ENT_QUOTES));
424 422                     $formula = str_replace('[DisplayId]', $displayId, $formula);
425 423
426 424                     $heading = str_replace('[DisplayGeoLocation]',
    $displayGeoLocation, $formula) . ' AS `' . $column->heading . '`';
    @@ -438,7 +436,7 @@ public function getData($filterBy = [], $options = [],
    $extraParams = [])
438 436             if ($filter != '') {
439 437                 // Support display filtering.
440 438                 $filter = str_replace('[DisplayId]', $displayId, $filter);
441     -               $filter = str_replace($this->blackList, '', $filter);
    439 +               $filter = Sql::cleanup($filter);
442 440
443 441                 $body .= ' AND ' . $filter;
444 442             }
```

lib/Helper/Sql.php

```
    @@ -0,0 +1,117 @@
1   + <?php
2   + /*
3   +  * Copyright (C) 2026 Xibo Signage Ltd
4   +  *
5   +  * Xibo - Digital Signage - http://www.xibo.org.uk
6   +  *
7   +  * This file is part of Xibo.
8   +  *
9   +  * Xibo is free software: you can redistribute it and/or modify
10  +  * it under the terms of the GNU Affero General Public License as published by
```

```
+  * the Free Software Foundation, either version 3 of the License, or
+  * any later version.
+  *
+  * Xibo is distributed in the hope that it will be useful,
+  * but WITHOUT ANY WARRANTY; without even the implied warranty of
+  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+  * GNU Affero General Public License for more details.
+  *
+  * You should have received a copy of the GNU Affero General Public License
+  * along with Xibo.  If not, see <http://www.gnu.org/licenses/>.
+  */
+
+ namespace Xibo\Helper;
+
+ /**
+  * SQL definitions
+  */
+ class Sql
+ {
+     /**
+      * Cleanup SQL (Maximum Paranoia for Legacy Code)
+      * @param string $sql the SQL to clean
+      * @param int $total the total number of replacements
+      * @return string
+      */
+     public static function cleanup($sql, &$total = 0)
+     {
+         $disallowedKeywords = [
+             ';', '@@', // Reduced symbols, handling comments via regex now
+             'INSERT', 'UPDATE', 'SELECT', 'FROM', 'WHERE', 'DELETE',
  'TRUNCATE',
+             'TABLE', 'ALTER', 'GRANT', 'REVOKE', 'CREATE', 'DROP', 'UNION',
+             'HAVING', 'GROUP', 'INTO', 'OUTFILE', 'DUMPFILE', 'PROCEDURE',
+             'SLEEP', 'BENCHMARK', 'INFORMATION_SCHEMA', 'LOAD_FILE', 'LOCK',
+             'EXECUTE', 'PREPARE', 'DEALLOCATE', 'SHOW', 'DESCRIBE', 'EXPLAIN',
+             'CALL', 'HANDLER', 'RENAME', 'SHUTDOWN', 'SET', 'USE', 'FLUSH',
+             'KILL', 'OPTIMIZE', 'REPAIR', 'ANALYZE', 'CHECK', 'CHECKSUM',
+             'GET_LOCK', 'RELEASE_LOCK', 'IS_FREE_LOCK', 'IS_USED_LOCK',
+             'MASTER_POS_WAIT', 'PASSWORD', 'USER', 'SYSTEM_USER',
  'SESSION_USER',
```

```
49  +             'CURRENT_USER', 'DATABASE', 'SCHEMA', 'VERSION', 'CONNECTION_ID',
50  +             'LAST_INSERT_ID', 'ROW_COUNT', 'FOUND_ROWS', 'LOAD_XML',
      'NAME_CONST',
51  +             'DO', 'EXTRACTVALUE', 'UPDATEXML', 'XMLTYPE', 'DBMS_PIPE',
      'PG_SLEEP',
52  +             // Added String Builders & Encoders
53  +             // we have specific use cases for 'CONCAT', so we keept that
54  +             'CONCAT_WS', 'CHAR', 'UNHEX', 'HEX', 'ASCII', 'BIN', 'ORD',
      'BASE64'
55  +         ];
56  +
57  +         // 1. EXTRACT AND PROTECT STRING LITERALS (Preserve user data)
58  +         $strings = [];
59  +         $placeholderPrefix = '__SQL_STR_';
60  +         $stringPattern = '/(\'(?:\\\\.|[^\'\\\\])*\'|"(?:\\\\.|[^"\\\\])*")/';
61  +
62  +         $sqlCleaned = preg_replace_callback($stringPattern, function ($matches)
      use (&$strings, $placeholderPrefix) {
63  +             $id = count($strings);
64  +             $strings[] = $matches[0];
65  +             return $placeholderPrefix . $id . '__';
66  +         }, $sql);
67  +
68  +         // 2. STRIP COMMENTS & ENCODINGS (Before keyword checks)
69  +         $preCleanupCount = 0;
70  +         $preCleanupPatterns = [
71  +             '/(?:\/\*.*?\*\/|--[ \t].*?(?:\n|$)|#[^\n]*?(?:\n|$))/s', //
      Standard & Executable Comments
72  +             '/\b0x[0-9a-fA-F]+\b/',                                    // Hex
      literals (e.g., 0x7e)
73  +             '/\bb\'[01]+\'/i'                                          // Binary
      literals
74  +         ];
75  +         $sqlCleaned = preg_replace($preCleanupPatterns, '', $sqlCleaned, -1,
      $preCleanupCount);
76  +         $total += $preCleanupCount;
77  +
78  +         // 3. PREPARE KEYWORD PATTERNS
79  +         $wordKeywords = [];
80  +         $symbolKeywords = [];
```

```
81  +
82  +         foreach ($disallowedKeywords as $keyword) {
83  +             if (ctype_alnum(str_replace('_', '', $keyword))) {
84  +                 $wordKeywords[] = preg_quote($keyword, '/');
85  +             } else {
86  +                 $symbolKeywords[] = $keyword;
87  +             }
88  +         }
89  +
90  +         $wordPattern = empty($wordKeywords) ? null : '/\b(' . implode('|',
    $wordKeywords) . ')\b/i';
91  +
92  +         // 4. RECURSIVE CLEANUP
93  +         $count = 0;
94  +         do {
95  +             $symbolCount = 0;
96  +             $wordCount = 0;
97  +
98  +             $sqlCleaned = str_ireplace($symbolKeywords, '', $sqlCleaned,
    $symbolCount);
99  +
100 +             if ($wordPattern) {
101 +                 $sqlCleaned = preg_replace($wordPattern, '', $sqlCleaned, -1,
    $wordCount);
102 +             }
103 +
104 +             $count = $symbolCount + $wordCount;
105 +             $total += $count;
106 +         } while ($count > 0);
107 +
108 +         // 5. RESTORE STRING LITERALS
109 +         if (!empty($strings)) {
110 +             foreach ($strings as $id => $originalString) {
111 +                 $sqlCleaned = str_replace($placeholderPrefix . $id . '__',
    $originalString, $sqlCleaned);
112 +             }
113 +         }
114 +
115 +         return trim($sqlCleaned);
116 +     }
```

```
117 + }
```

Comments 0

Please sign in to comment.