

xwiki / xwiki-platform Public

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

Commit 3c8a2ec

michitux committed on Dec 2, 2025 · ✓ 1/1

XWIKI-23472: URL generation in the changes view is broken

- * Refactor changesdoc.vm to fix URL handling.
- * Add unit tests for changes template

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

1 parent 6d7447c commit 3c8a2ec

2 files changed +177 -8 lines changed

↑ Top ⚙️

🔍 Filter files...

- ✖️ xwiki-platform-core/xwiki-platform-web/xwiki-platform-web-templates/src
 - ✖️ main/resources/templates
 - changesdoc.vm
 - ✖️ test/java/org/xwiki/web
 - ChangesPageTest.java

2 files changed +177 -8 lines changed

🔍 Search within code ⚙️

.../src/main/resources/templates/changesdoc.vm

```

@@ -23,44 +23,52 @@
23 23  ##
24 24
25 25  #if ("${previousOrigdocVersion}" != '')
26 -   #set ($changesLink = ${xwiki.getURL($tdoc, 'view',
      "viewer=changes&rev1=${previousOrigdocVersion}&rev2=${rev2}"))
26 +   #set ($changesLink = ${xwiki.getURL($tdoc, 'view',

```

```

27 +   $escapetool.url({'viewer': 'changes', 'rev1': $previousOrigdocVersion,
#set ($changesTitle =
$escapetool.xml($services.localization.render('core.viewers.diff.previousVersion
.origin', [${previousOrigdocVersion}])))
28 29   #set ($previousOrigdocLink = "<a class='btn btn-default changes-arrow'
href='$changesLink' title='$changesTitle'>$services.icon.renderHTML('left')
</a>")
29 30 #end
30 31 #if ("${nextOrigdocVersion}" != '')
31 -   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',
"viewer=changes&rev1=${nextOrigdocVersion}&rev2=${rev2}")})
32 +   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',
33 +   $escapetool.url({'viewer': 'changes', 'rev1': $nextOrigdocVersion, 'rev2':
$rev2})})
32 34   #set ($changesTitle =
$escapetool.xml($services.localization.render('core.viewers.diff.nextVersion.ori
gin', [${nextOrigdocVersion}])))
33 35   #set ($nextOrigdocLink = "<a class='btn btn-default changes-arrow'
href='$changesLink' title='$changesTitle'>$services.icon.renderHTML('right')
</a>")
34 36 #end
35 37 #if ("${previousNewdocVersion}" != '')
36 -   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',
"viewer=changes&rev1=${rev1}&rev2=${previousNewdocVersion}")})
38 +   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',
39 +   $escapetool.url({'viewer': 'changes', 'rev1': $rev1, 'rev2':
$previousNewdocVersion})})
37 40   #set ($changesTitle =
$escapetool.xml($services.localization.render('core.viewers.diff.previousVersion
.target', [${previousNewdocVersion}])))
38 41   #set ($previousNewdocLink = "<a class='btn btn-default changes-arrow'
href='$changesLink' title='$changesTitle'>$services.icon.renderHTML('left')
</a>")
39 42 #end
40 43 #if ("${nextNewdocVersion}" != '')
41 -   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',
"viewer=changes&rev1=${rev1}&rev2=${nextNewdocVersion}")})
44 +   #set ($changesLink = ${xwiki.getUrl($tdoc, 'view',

```

```

45 +   $escapetool.url({'viewer': 'changes', 'rev1': $rev1, 'rev2':
    $nextNewdocVersion}}))}}
42 46   #set ($changesTitle =
    $escapetool.xml($services.localization.render('core.viewers.diff.nextVersion.tar
    get', [${nextNewdocVersion}])))
43 47   #set ($nextNewdocLink = "<a class='btn btn-default changes-arrow'
    href='$changesLink' title='$changesTitle'>$services.icon.renderHTML('right')
    </a>")
44 48   #end
45 49   ## Display the previous change arrow only if there is a previous version of the
    original document and there is no extension version
46 50   #if ("${previousOrigdocVersion}" != '' && "${previousNewdocVersion}" != '')
47 51   ## Keep the reverse navigation order in case the original document version is
    greater than the version of the new document
48 52   #if ($origdoc.RCSVersion.isLessOrEqualThan($newdoc.RCSVersion))
49 -   #set ($previousLink = ${xwiki.getUrl($doc, 'view',
    "viewer=changes&rev1=${previousOrigdocVersion}&rev2=${rev1}")})
53 +   #set ($previousLink = ${xwiki.getUrl($doc, 'view',
54 +   $escapetool.url({'viewer': 'changes', 'rev1': $previousOrigdocVersion,
    'rev2': $rev1}}))
50 55   #set ($previousLabel =
    $escapetool.xml($services.localization.render('core.viewers.diff.change',
    [${previousOrigdocVersion,$rev1}]))
51 56   #else
52 -   #set ($previousLink = ${xwiki.getUrl($doc, 'view',
    "viewer=changes&rev1=${rev2}&rev2=${previousNewdocVersion}")})
57 +   #set ($previousLink = ${xwiki.getUrl($doc, 'view',
58 +   $escapetool.url({'viewer': 'changes', 'rev1': $rev2, 'rev2':
    $previousNewdocVersion}}))
53 59   #set ($previousLabel =
    $escapetool.xml($services.localization.render('core.viewers.diff.change',
    [${rev2, $previousNewdocVersion}]))
54 60   #end
55 61   #end
56 62   ## Display the next change arrow only if there is a next version of the new
    document and there is no extension version
57 63   #if ("${nextOrigdocVersion}" != '' && "${nextNewdocVersion}" != '')
58 64   ## Keep the reverse navigation order in case the original document version is
    greater than the version of the new document
59 65   #if ($origdoc.RCSVersion.isLessOrEqualThan($newdoc.RCSVersion))

```

60	-	<code>#set (\$nextLink = \${xwiki.getUrl(\$tdoc, 'view', "viewer=changes&rev1=\${rev2}&rev2=\${nextNewdocVersion}"))}</code>
66	+	<code>#set (\$nextLink = \${xwiki.getUrl(\$tdoc, 'view',</code>
67	+	<code>\$escapetool.url({'viewer': 'changes', 'rev1': \$rev2, 'rev2': \$nextNewdocVersion})))}</code>
61	68	<code>#set (\$nextLabel = \$escapetool.xml(\$services.localization.render('core.viewers.diff.change', [\$rev2, \$nextNewdocVersion])))</code>
62	69	<code>#else</code>
63	-	<code>#set (\$nextLink = \${xwiki.getUrl(\$tdoc, 'view', "viewer=changes&rev1=\${nextOrigdocVersion}&rev2=\${rev1}"))}</code>
70	+	<code>#set (\$nextLink = \${xwiki.getUrl(\$tdoc, 'view',</code>
71	+	<code>\$escapetool.url({'viewer': 'changes', 'rev1': \$nextOrigdocVersion, 'rev2': \$rev1})))}</code>
64	72	<code>#set (\$nextLabel = \$escapetool.xml(\$services.localization.render('core.viewers.diff.change', [\$nextOrigdocVersion, \$rev1])))</code>
65	73	<code>#end</code>
66	74	<code>#end</code>
		↓

<div style="display: flex; justify-content: space-between; align-items: center;"> ▼ ...est/java/org/xwiki/web/ChangesPageTest.java ⋮ </div>	<div style="display: flex; justify-content: space-between; align-items: center;"> ⋮ @@ -0,0 +1,161 @@ </div>
1	+ /*
2	+ * See the NOTICE file distributed with this work for additional
3	+ * information regarding copyright ownership.
4	+ *
5	+ * This is free software; you can redistribute it and/or modify it
6	+ * under the terms of the GNU Lesser General Public License as
7	+ * published by the Free Software Foundation; either version 2.1 of
8	+ * the License, or (at your option) any later version.
9	+ *
10	+ * This software is distributed in the hope that it will be useful,
11	+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
12	+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13	+ * Lesser General Public License for more details.
14	+ *
15	+ * You should have received a copy of the GNU Lesser General Public
16	+ * License along with this software; if not, write to the Free
17	+ * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA

```
18 + * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
19 + */
20 + package org.xwiki.web;
21 +
22 + import org.apache.commons.lang3.Strings;
23 + import org.junit.jupiter.api.BeforeEach;
24 + import org.junit.jupiter.api.Test;
25 + import org.xwiki.model.reference.DocumentReference;
26 + import org.xwiki.rendering.syntax.Syntax;
27 + import org.xwiki.template.TemplateManager;
28 + import org.xwiki.test.junit5.mockito.MockComponent;
29 + import org.xwiki.test.page.PageTest;
30 +
31 + import com.xpn.xwiki.criteria.impl.XWikiCriteriaServiceImpl;
32 + import com.xpn.xwiki.doc.DocumentRevisionProvider;
33 + import com.xpn.xwiki.doc.XWikiDocument;
34 +
35 + import static org.hamcrest.MatcherAssert.assertThat;
36 + import static org.hamcrest.Matchers.containsString;
37 + import static org.hamcrest.Matchers.not;
38 + import static org.junit.jupiter.api.Assertions.assertTrue;
39 + import static org.mockito.ArgumentMatchers.any;
40 + import static org.mockito.ArgumentMatchers.anyString;
41 + import static org.mockito.Mockito.when;
42 +
43 + /**
44 +  * Tests the {@code changes.vm} template.
45 +  *
46 +  * @version $Id$
47 +  */
48 + class ChangesPageTest extends PageTest
49 + {
50 +     private static final String CHANGES_VM = "changes.vm";
51 +
52 +     private static final String INVALID_REVISION = "xar:'>Invalid<'";
53 +
54 +     private static final String ENCODED_INVALID_REVISION =
55         "xar%3A%27%3EInvalid%3C%27";
```

```
56 +     private static final DocumentReference DOCUMENT_REFERENCE = new
    DocumentReference("xwiki", "TestSpace", "TestPage");
57 +
58 +     private TemplateManager templateManager;
59 +
60 +     @MockComponent
61 +     private DocumentRevisionProvider documentRevisionProvider;
62 +
63 +     @BeforeEach
64 +     void setUp() throws Exception
65 +     {
66 +         this.templateManager =
    this.oldcore.getMocker().getInstance(TemplateManager.class);
67 +
68 +         this.xwiki.setCriteriaService(new
    XWikiCriteriaServiceImpl(this.context));
69 +
70 +         // Create several revisions of the document for testing
71 +         createDocumentRevisions();
72 +
73 +         // Set up the current doc in the context so that $doc is bound in
    scripts
74 +         this.context.setDoc(this.xwiki.getDocument(DOCUMENT_REFERENCE,
    this.context));
75 +
76 +         // Get the mock database revision provider that was set up by
    MockitoOldcore.
77 +         DocumentRevisionProvider databaseRevisionProvider =
78 +
    this.oldcore.getMocker().getInstance(DocumentRevisionProvider.class,
    "database");
79 +
80 +         // Mock the default revision provider to handle xar: revision numbers
    for the test document.
81 +
    when(this.documentRevisionProvider.getRevision(any(DocumentReference.class),
    anyString()))
82 +         .thenAnswer(invocation -> {
83 +             DocumentReference reference = invocation.getArgument(0);
84 +             String revision = invocation.getArgument(1);
```

```
85 +         if (Strings.CS.startsWith(revision, "xar:")
86 +             && DOCUMENT_REFERENCE.equals(reference.withoutLocale())) {
87 +             return this.context.getDoc();
88 +         }
89 +         return databaseRevisionProvider.getRevision(reference,
revision);
90 +     });
91 + }
92 +
93 + private void createDocumentRevisions() throws Exception
94 + {
95 +     XWikiDocument document = new XWikiDocument(DOCUMENT_REFERENCE);
96 +     document.setSyntax(Syntax.XWIKI_2_1);
97 +     document.setTitle("Test Page for Changes");
98 +
99 +     // Revision 1.1.
100 +     document.setContent("Initial content");
101 +     document.setComment("Initial version");
102 +     this.xwiki.saveDocument(document, this.context);
103 +
104 +     // Revision 2.1.
105 +     document.setContent("Initial content\nSecond line");
106 +     document.setComment("Added second line");
107 +     this.xwiki.saveDocument(document, this.context);
108 +
109 +     // Revision 3.1.
110 +     document.setContent("Initial content\nSecond line\nThird line");
111 +     document.setComment("Added third line");
112 +     this.xwiki.saveDocument(document, this.context);
113 +
114 +     // Revision 4.1.
115 +     document.setContent("Initial content\nSecond line\nFourth line");
116 +     document.setComment("Added fourth line");
117 +     this.xwiki.saveDocument(document, this.context);
118 + }
119 +
120 + @Test
121 + void renderChangesTemplateWithBasicParameters() throws Exception
122 + {
123 +     this.stubRequest.put("rev1", "1.1");
```

```
124 +         this.stubRequest.put("rev2", "2.1");
125 +
126 +         String result = this.templateManager.render(CHANGES_VM);
127 +
128 +         assertTrue(result.contains("document-title"));
129 +         assertTrue(result.contains("Test Page for Changes"));
130 +     }
131 +
132 +     @Test
133 +     void renderChangesTemplateWithInvalidRev2Parameters() throws Exception
134 +     {
135 +         // Set up request parameters for changes view with invalid revision
136 +         // numbers
137 +         this.stubRequest.put("rev1", "2.1");
138 +         this.stubRequest.put("rev2", INVALID_REVISION);
139 +
140 +         String result = this.templateManager.render(CHANGES_VM);
141 +
142 +         // Verify that the revision has been sanitized.
143 +         assertTrue(result, not(containsString(INVALID_REVISION)));
144 +         // Verify that the URL-escaped revision is present in the result.
145 +         assertTrue(result, containsString(ENCODED_INVALID_REVISION));
146 +     }
147 +
148 +     @Test
149 +     void renderChangesTemplateWithInvalidRev1Parameters() throws Exception
150 +     {
151 +         // Set up request parameters for changes view with invalid revision
152 +         // numbers
153 +         this.stubRequest.put("rev1", INVALID_REVISION);
154 +         this.stubRequest.put("rev2", "3.1");
155 +
156 +         String result = this.templateManager.render(CHANGES_VM);
157 +
158 +         // Verify that the revision has been sanitized.
159 +         assertTrue(result, not(containsString(INVALID_REVISION)));
160 +         // Verify that the URL-escaped revision is present in the result.
161 +         assertTrue(result, containsString(ENCODED_INVALID_REVISION));
162 +     }
163 + }
```

Comments 0



Please [sign in](#) to comment.