xwiki / **xwiki-commons**

[◉ Watch   26]   [★ Star   54]   [⑂ Fork   83]

<> Code   ⏻ Pull requests 4   ⊞ Projects 0   ┾ Insights

Merge branch 'bug-XCOMMONS-1342'   [Browse files]

ᛘ master   ◇ xwiki-commons-10.8   ⋯   ◇ xwiki-commons-10.0

👤 gdelhumeau committed on Jan 18   2 parents a0b25ca + 9950086   commit d3101aed5d0b3df40e9ac9cf1a40af74262d5f75

⌖ Showing **2 changed files** with **71 additions** and **9 deletions**.   [Unified] [Split]

44 ▉▉▉▉▁▁ ...diff/xwiki-commons-diff-api/src/main/java/org/xwiki/diff/internal/DefaultDiffManager.java   [View] [⌄]

```
127  127                @@ -127,6 +127,16 @@
128  128                            if (patchCurrent.isEmpty()) {
129  129                                mergeResult.setMerged(next);
     130  +                          } else if (isFullyModified(commonAncestor, patchCurrent)) {
     131  +                              // If current is completely modified compared to the common ancestor we assume any change in next is
     132  +                              // a conflict
     133  +                              // ... except if the current content is identical to the next one!
     134  +                              if (!current.equals(next)) {
     135  +                                  Delta<E> deltaNext = nextElement(patchNext);
     136  +                                  Delta<E> deltaCurrent = nextElement(patchCurrent);
     137  +                                  logConflict(mergeResult, deltaCurrent, deltaNext);
     138  +                              }
     139  +                              mergeResult.setMerged(fallback(commonAncestor, next, current, configuration));
130  140                            } else {
131  141                                merge(mergeResult, commonAncestor, patchNext, patchCurrent, configuration);
132  142                            }
```

```
135  145                        return mergeResult;
136  146                    }
137  147
     148  +    private <E> List<E> fallback(List<E> commonAncestor, List<E> next, List<E> current,
     149  +        MergeConfiguration<E> configuration)
     150  +    {
     151  +        Version fallbackVersion;
     152  +        if (configuration != null) {
     153  +            fallbackVersion = configuration.getFallbackOnConflict();
     154  +        } else {
     155  +            fallbackVersion = Version.CURRENT;
     156  +        }
     157  +
     158  +        switch (fallbackVersion) {
     159  +            case NEXT:
     160  +                return next;
     161  +            case PREVIOUS:
     162  +                return commonAncestor;
     163  +            default:
     164  +                return current;
     165  +        }
     166  +    }
     167  +
138  168        private <E> int fallback(List<E> commonAncestor, Delta<E> deltaNext, Delta<E> deltaCurrent, List<E> merged,
139  -            int currentIndex, MergeConfiguration<E> configuration)
     169  +        int currentIndex, MergeConfiguration<E> configuration)
140  170        {
141  171            int newIndex = currentIndex;
142  172
```

```
357  387        {
358  388            return delta != null && delta.getPrevious().getIndex() == index;
359  389        }
     390  +
     391  +    /**
     392  +     * Check if the content is completely different between the ancestor and the current version
     393  +     *
     394  +     * @param <E> the type of compared elements
     395  +     * @param commonAncestor previous version
     396  +     * @param patchCurrent patch to the current version
     397  +     * @return either or not the user has changed everything
     398  +     */
     399  +    private <E> boolean isFullyModified(List commonAncestor, Patch<E> patchCurrent) {
     400  +        return patchCurrent.size() == 1 && commonAncestor.size() == patchCurrent.get(0).getPrevious().size();
     401  +    }
360  402    }
```

36 ▉▉▉▉▉▁ .../xwiki-commons-diff-api/src/test/java/org/xwiki/diff/internal/DefaultDiffManagerTest.java   [View] [⌄]

```
264  264            Assert.assertEquals(toCharacters("aijb"), result.getMerged());
265  265
266  266            result =
267  -                this.mocker.getComponentUnderTest().merge(toCharacters(""), toCharacters("ab"), toCharacters("abc"), null);
     267  +                this.mocker.getComponentUnderTest().merge(toCharacters("d"), toCharacters("ab d"), toCharacters("abc d"), null
268  268
269  269            Assert.assertEquals(0, result.getLog().getLogs(LogLevel.ERROR).size());
270  -            Assert.assertEquals(toCharacters("abc"), result.getMerged());
     270  +            Assert.assertEquals(toCharacters("abc d"), result.getMerged());
271  271
272  272            result =
273  -                this.mocker.getComponentUnderTest().merge(toCharacters(""), toCharacters("abc"), toCharacters("ab"), null);
     273  +                this.mocker.getComponentUnderTest().merge(toCharacters("d"), toCharacters("abc d"), toCharacters("ab d"), null
274  274
275  275            Assert.assertEquals(0, result.getLog().getLogs(LogLevel.ERROR).size());
276  -            Assert.assertEquals(toCharacters("abc"), result.getMerged());
     276  +            Assert.assertEquals(toCharacters("abc d"), result.getMerged());
277  277
278  278            result =
279  -                this.mocker.getComponentUnderTest().merge(toCharacters(""), toCharacters("bc"), toCharacters("abc"), null);
     279  +                this.mocker.getComponentUnderTest().merge(toCharacters("d"), toCharacters("bcd"), toCharacters("abcd"), null);
280  280
281  281            Assert.assertEquals(0, result.getLog().getLogs(LogLevel.ERROR).size());
282  -            Assert.assertEquals(toCharacters("abc"), result.getMerged());
     282  +            Assert.assertEquals(toCharacters("abcd"), result.getMerged());
283  283
284  284            result =
285  -                this.mocker.getComponentUnderTest().merge(toCharacters(""), toCharacters("abc"), toCharacters("bc"), null);
     285  +                this.mocker.getComponentUnderTest().merge(toCharacters("d"), toCharacters("abcd"), toCharacters("bcd"), null);
286  286
287  287            Assert.assertEquals(0, result.getLog().getLogs(LogLevel.ERROR).size());
288  -            Assert.assertEquals(toCharacters("abc"), result.getMerged());
     288  +            Assert.assertEquals(toCharacters("abcd"), result.getMerged());
289  289
290  290            // Misc
291  291
```

```
354  354            Assert.assertEquals(1, result.getLog().getLogs(LogLevel.ERROR).size());
355  355            Assert.assertEquals(toCharacters("b"), result.getMerged());
356  356        }
     357  +
     358  +    @Test
     359  +    public void testMergeWhenUserHasChangedAllContent() throws Exception
     360  +    {
     361  +        MergeResult<String> result;
     362  +
     363  +        // Test 1: All content has changed between previous and current
     364  +        result = mocker.getComponentUnderTest().merge(Arrays.asList("Line 1", "Line 2", "Line 3"),
     365  +            Arrays.asList("Line 1", "Line 2 modified", "Line 3", "Line 4 Added"),
     366  +            Arrays.asList("New content", "That is completely different"), null);
     367  +
     368  +        Assert.assertEquals(Arrays.asList("New content", "That is completely different"), result.getMerged());
     369  +
     370  +        // Test 2: All content has been deleted between previous and current
     371  +        result = mocker.getComponentUnderTest().merge(Arrays.asList("Line 1", "Line 2", "Line 3"),
     372  +            Arrays.asList("Line 1", "Line 2 modified", "Line 3", "Line 4 Added"),
     373  +            Collections.emptyList(), null);
     374  +
     375  +        Assert.assertEquals(Collections.emptyList(), result.getMerged());
     376  +    }
357  377    }
```

0 comments on commit d3101ae

Please sign in to comment.