

Quote.php

```
1. <?php
2.
3. namespace dokuwiki\Parsing\Handler;
4.
5. class Quote extends AbstractRewriter
6. {
7.     protected $quoteCalls = array();
8.
9.     /** @inheritdoc */
10.    public function finalise()
11.    {
12.        $last_call = end($this->calls);
13.        $this->writeCall(array('quote_end',array(),
14.        $last_call[2]));
15.
16.        $this->process();
17.        $this->callWriter->finalise();
18.        unset($this->callWriter);
19.    }
20.
21.    /** @inheritdoc */
22.    public function process()
23.    {
24.        $quoteDepth = 1;
25.
26.        foreach ($this->calls as $call) {
27.            switch ($call[0]) {
28.
29.                /** @noinspection
30.                 *PhpMissingBreakStatementInspection */
31.                case 'quote_start':
32.                    $this->quoteCalls[] =
33.                    array('quote_open',array(),$call[2]);
34.                    // fallthrough
35.                    case 'quote_newline':
36.                        $quoteLength = $this->getDepth($call[1][0]);
37.
38.                        if ($quoteLength > $quoteDepth) {
39.                            $quoteDiff = $quoteLength - $quoteDepth;
40.                            for ($i = 1; $i <= $quoteDiff; $i++) {
41.                                $this->quoteCalls[] =
42.                                array('quote_open',array(),$call[2]);
43.                            }
44.                        } elseif ($quoteLength < $quoteDepth) {
45.                            $quoteDiff = $quoteDepth - $quoteLength;
46.                            for ($i = 1; $i <= $quoteDiff; $i++) {
47.                                $this->quoteCalls[] =
```

```
array('quote_close',array(),$call[2]);
45.         }
46.     } else {
47.         if ($call[0] != 'quote_start')
$this->quoteCalls[] = array('linebreak',array(),$call[2]);
48.     }
49.
50.         $quoteDepth = $quoteLength;
51.
52.         break;
53.
54.     case 'quote_end':
55.         if ($quoteDepth > 1) {
56.             $quoteDiff = $quoteDepth - 1;
57.             for ($i = 1; $i <= $quoteDiff; $i++) {
58.                 $this->quoteCalls[] =
array('quote_close',array(),$call[2]);
59.             }
60.         }
61.
62.         $this->quoteCalls[] =
array('quote_close',array(),$call[2]);
63.
64.         $this->callWriter->writeCalls($this->quoteCalls);
65.         break;
66.
67.     default:
68.         $this->quoteCalls[] = $call;
69.         break;
70.     }
71. }
72.
73.     return $this->callWriter;
74. }
75.
76. /**
77.  * @param string $marker
78.  * @return int
79.  */
80. protected function getDepth($marker)
81. {
82.     preg_match('/>{1,}/', $marker, $matches);
83.     $quoteLength = strlen($matches[0]);
84.     return $quoteLength;
85. }
86. }
```

From:

<https://wwoss.ru/> - **worldwide open-source software**

Permanent link:

<https://wwoss.ru/doku.php?id=wiki:xref:dokuwiki:inc:parsing:handler:quote.php>

Last update: **2025/01/16 22:01**

