

Parser.php

```
1. <?php
2.
3. namespace dokuwiki\Parsing;
4.
5. use Doku_Handler;
6. use dokuwiki\Parsing\Lexer\Lexer;
7. use dokuwiki\Parsing\ParserMode\Base;
8. use dokuwiki\Parsing\ParserMode\ModeInterface;
9.
10. /**
11.  * Sets up the Lexer with modes and points it to the Handler
12.  * For an intro to the Lexer see: wiki:parser
13.  */
14. class Parser {
15.
16.     /** @var Doku_Handler */
17.     protected $handler;
18.
19.     /** @var Lexer $lexer */
20.     protected $lexer;
21.
22.     /** @var ModeInterface[] $modes */
23.     protected $modes = array();
24.
25.     /** @var bool mode connections may only be set up once */
26.     protected $connected = false;
27.
28.     /**
29.      * dokuwiki\Parsing\Doku_Parser constructor.
30.      *
31.      * @param Doku_Handler $handler
32.      */
33.     public function __construct(Doku_Handler $handler) {
34.         $this->handler = $handler;
35.     }
36.
37.     /**
38.      * Adds the base mode and initialized the lexer
39.      *
40.      * @param Base $BaseMode
41.      */
42.     protected function addBaseMode($BaseMode) {
43.         $this->modes['base'] = $BaseMode;
44.         if(!$this->lexer) {
45.             $this->lexer = new Lexer($this->handler, 'base',
true);
46.         }
47.         $this->modes['base']->Lexer = $this->lexer;
```

```
48.     }
49.
50.     /**
51.      * Add a new syntax element (mode) to the parser
52.      *
53.      * PHP preserves order of associative elements
54.      * Mode sequence is important
55.      *
56.      * @param string $name
57.      * @param ModeInterface $Mode
58.      */
59.     public function addMode($name, ModeInterface $Mode) {
60.         if(!isset($this->modes['base'])) {
61.             $this->addBaseMode(new Base());
62.         }
63.         $Mode->Lexer = $this->lexer; // FIXME should be done by
setter
64.         $this->modes[$name] = $Mode;
65.     }
66.
67.     /**
68.      * Connect all modes with each other
69.      *
70.      * This is the last step before actually parsing.
71.      */
72.     protected function connectModes() {
73.
74.         if($this->connected) {
75.             return;
76.         }
77.
78.         foreach(array_keys($this->modes) as $mode) {
79.             // Base isn't connected to anything
80.             if($mode == 'base') {
81.                 continue;
82.             }
83.             $this->modes[$mode]->preConnect();
84.
85.             foreach(array_keys($this->modes) as $cm) {
86.
87.                 if($this->modes[$cm]->accepts($mode)) {
88.                     $this->modes[$mode]->connectTo($cm);
89.                 }
90.
91.             }
92.
93.             $this->modes[$mode]->postConnect();
94.         }
95.
96.         $this->connected = true;
```

```
97.     }
98.
99.     /**
100.      * Parses wiki syntax to instructions
101.      *
102.      * @param string $doc the wiki syntax text
103.      * @return array instructions
104.      */
105.     public function parse($doc) {
106.         $this->connectModes();
107.         // Normalize CRs and pad doc
108.         $doc = "\n" . str_replace("\r\n", "\n", $doc) . "\n";
109.         $this->lexer->parse($doc);
110.
111.         if (!method_exists($this->handler, 'finalize')) {
112.             /** @deprecated 2019-10 we have a legacy handler from
113.              * a plugin, assume legacy _finalize exists */
114.
115.             \dokuwiki\Debug\DebugHelper::dbgCustomDeprecationEvent(
116.                 'finalize()',
117.                 get_class($this->handler) . '::_finalize()',
118.                 __METHOD__,
119.                 __FILE__,
120.                 __LINE__
121.             );
122.             $this->handler->_finalize();
123.         } else {
124.             $this->handler->finalize();
125.         }
126.         return $this->handler->calls;
127.     }
128. }
```

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

Permanent link:
<https://wwoss.ru/doku.php?id=wiki:xref:dokuwiki:inc:parsing:parser.php>

Last update: **2025/01/16 22:58**

