

Parser.php

```
<?php

namespace dokuwiki\Parsing;

use Doku_Handler;
use dokuwiki\Parsing\Lexer\Lexer;
use dokuwiki\Parsing\ParserMode\Base;
use dokuwiki\Parsing\ParserMode\ModeInterface;

/**
 * Sets up the Lexer with modes and points it to the Handler
 * For an intro to the Lexer see: wiki:parser
 */
class Parser {

    /** @var Doku_Handler */
    protected $handler;

    /** @var Lexer $lexer */
    protected $lexer;

    /** @var ModeInterface[] $modes */
    protected $modes = array();

    /** @var bool mode connections may only be set up once */
    protected $connected = false;

    /**
     * dokuwiki\Parsing\Doku_Parser constructor.
     *
     * @param Doku_Handler $handler
     */
    public function __construct(Doku_Handler $handler) {
        $this->handler = $handler;
    }

    /**
     * Adds the base mode and initialized the lexer
     *
     * @param Base $BaseMode
     */
    protected function addBaseMode($BaseMode) {
        $this->modes['base'] = $BaseMode;
        if(!$this->lexer) {
            $this->lexer = new Lexer($this->handler, 'base', true);
        }
        $this->modes['base']->Lexer = $this->lexer;
    }
}
```

```
/**
 * Add a new syntax element (mode) to the parser
 *
 * PHP preserves order of associative elements
 * Mode sequence is important
 *
 * @param string $name
 * @param ModeInterface $Mode
 */
public function addMode($name, ModeInterface $Mode) {
    if(!isset($this->modes['base'])) {
        $this->addBaseMode(new Base());
    }
    $Mode->Lexer = $this->lexer; // FIXME should be done by setter
    $this->modes[$name] = $Mode;
}

/**
 * Connect all modes with each other
 *
 * This is the last step before actually parsing.
 */
protected function connectModes() {

    if($this->connected) {
        return;
    }

    foreach(array_keys($this->modes) as $mode) {
        // Base isn't connected to anything
        if($mode == 'base') {
            continue;
        }
        $this->modes[$mode]->preConnect();

        foreach(array_keys($this->modes) as $cm) {

            if($this->modes[$cm]->accepts($mode)) {
                $this->modes[$mode]->connectTo($cm);
            }

        }

        $this->modes[$mode]->postConnect();
    }

    $this->connected = true;
}

/**
```

```
* Parses wiki syntax to instructions
*
* @param string $doc the wiki syntax text
* @return array instructions
*/
public function parse($doc) {
    $this->connectModes();
    // Normalize CRs and pad doc
    $doc = "\n" . str_replace("\r\n", "\n", $doc) . "\n";
    $this->lexer->parse($doc);

    if (!method_exists($this->handler, 'finalize')) {
        /** @deprecated 2019-10 we have a legacy handler from a
        plugin, assume legacy _finalize exists */

        \dokuwiki\Debug\DebugHelper::dbgCustomDeprecationEvent(
            'finalize()',
            get_class($this->handler) . '::_finalize()',
            __METHOD__,
            __FILE__,
            __LINE__
        );
        $this->handler->_finalize();
    } else {
        $this->handler->finalize();
    }
    return $this->handler->calls;
}
}
```

From:
<http://synoinstall-gqctx9n8ug2b3eq1.direct.quickconnect.to/> - worldwide open-source software

Permanent link:
<http://synoinstall-gqctx9n8ug2b3eq1.direct.quickconnect.to/doku.php?id=wiki:xref:dokuwiki:inc:parsing:parser.php&rev=1737019524>

Last update: 2025/01/16 12:25

