

Скелет синтаксического плагина

[syntax.php](#)

```
<?php
/**
 * Плагин Skeleton: отображает «Hello World!»
 *
 * Синтаксис: <TEST> – будет заменен на «Hello World!»
 *
 * @license GPL 2 (http://www.gnu.org/licenses/gpl.html)
 * @автор Кристофер Смит <chris@jalakai.co.uk>
 */

/**
 * Все плагины DokuWiki для расширения механизма парсера/рендеринга
 * необходимо унаследовать от этого класса
 */
class syntax_plugin_test extends DokuWiki_Syntax_Plugin {

    /**
     * Получите тип синтаксиса, который определяет этот плагин.
     *
     * @param нет
     * @return String <tt>'substition'</tt> (т. е. 'подстановка').
     * @общественность
     * @статичный
     */
    function getType(){
        return 'substition';
    }

    /**
     * Какой синтаксис мы допускаем (необязательно)
     */
    //    function getAllowedTypes() {
    //        return array();
    //    }

    /**
     * Определите, как этот плагин обрабатывается в отношении абзацев.
     *
     * <p>
     * Этот метод важен для правильного вложения XHTML. Он возвращает
     * одно из следующих значений:
     * </p>
    
```

```

* <дл>
  * <dt>normal</dt><dd>Плагин можно использовать внутри абзацев.</dd>
  * <dt>блок</dt><dd>Открытые абзацы необходимо закрыть перед
* вывод плагина.</dd>
  * <dt>стек</dt><dd>Особый случай: плагин обворачивает другие абзацы.</dd>
  * </dl>
* @параметр нет
  * @return String <tt>'блок'</tt>.
* @общественность
* @статичный
  */
//    function getPType(){
//        return 'normal';
//    }

/**
* Где сортировать?
*
* @param none
* @return Integer <tt>6</tt>.
* @public
* @static
*/
function getSort(){
    return 999;
}

/**
* Подключить шаблон поиска к лексеру.
*
* @param $aMode String Требуемый режим визуализации.
* @return none
* @public
* @see render()
*/
function connectTo($mode) {
    $this->Lexer->addSpecialPattern('<TEST>', $mode, 'plugin_test');
//    $this->Lexer->addEntryPattern('<TEST>', $mode, 'plugin_test');
}

//    function postConnect() {
//        $this->Lexer->addExitPattern('</TEST>', 'plugin_test');
//    }

/**
* Обработчик для подготовки сопоставленных данных для процесса рендеринга.
*
* <p>
* Параметр <tt>$aState</tt> задает тип шаблона

```

```

, * который вызвал вызов этого метода:
 * </p>
 * <dl>
 * <dt>DOKU_LEXER_ENTER</dt>
 * <dd>шаблон , установленный функцией <tt>addEntryPattern()</tt></dd>
*
<dt>DOKU_LEXER_MATCHED </dt> * <dd>
шаблон, установленный функцией <tt>addPattern()</tt></dd>
* <dt>DOKU_LEXER_EXIT </dt>
* <dd>шаблон, установленный функцией <tt>addExitPattern()</tt> </dd>
* <dt>DOKU_LEXER_SPECIAL</dt>
* <dd>шаблон, установленный функцией
<tt>addSpecialPattern()</tt></dd>
* <dt>DOKU_LEXER_UNMATCHED</dt>
* <dd>обычный текст, встречающийся в режим синтаксиса плагина
, * который не соответствует ни одному шаблону.</dd>
* </dl>
* @param $aMatch String Текст, соответствующий шаблонам.
* @param $aState Integer Состояние лексера для сопоставления.
* @param $aPos Integer Позиция символа сопоставленного текста.
* @param $aHandler Object Ссылка на объект Doku_Handler.
* @return Integer Текущее состояние лексера для сопоставления.
* @public
* @see render()
* @static
*/
function handle($match, $state, $pos, Doku_Handler $handler){
    switch ($state) {
        case DOKU_LEXER_ENTER :
            break;
        case DOKU_LEXER_MATCHED :
            break;
        case DOKU_LEXER_UNMATCHED :
            break;
        case DOKU_LEXER_EXIT :
            break;
        case DOKU_LEXER_SPECIAL :
            break;
    }
    return array();
}

/**
* Обработка фактического создания выходных данных.
*
* <p>
* Метод проверяет заданный <tt>$aFormat</tt> и возвращает
* <tt>FALSE</tt>, если формат не поддерживается. <tt>$aRenderer</tt>
* содержит ссылку на объект рендерера, который в данный момент
* обрабатывает рендеринг. Содержимое <tt>$aData</tt> является
* возвращаемым значением метода <tt>handle()</tt>.

```

```
* </p>
* @param $aFormat String Формат выходных данных для генерации.
* @param $aRenderer Object Ссылка на объект рендерера.
* @param $aData Array Данные, созданные методом <tt>handle()</tt>
*.
* @return Boolean <tt>TRUE</tt> при успешном рендеринге или
* <tt>FALSE</tt> в противном случае.
* @public
* @see handle()
*/
function render($mode, Doku_Renderer $renderer, $data) {
    if($mode == 'xhtml'){
        $renderer->doc .= "Hello World!"; // ptype =
'normal'
    }
    else
        $renderer->doc .= "<p>Hello World!</p>"; // ptype =
'block'
    return true;
}
return false;
}
```

Примечание:

Имя этого плагина — `test`. Это видно по имени класса: `syntax_plugin_test` — все, что следует далее, `syntax_plugin_` воспринимается как имя плагина.

Поскольку имя плагина — `test`, его нужно сохранить в `lib/plugins/test` чтобы DokuWiki автоматически распознал его. Его нужно сохранить как `syntax.php`

Дополнения и Файлы

см. также

- [plugin_programming_tips](#)
- [Ссылки на источник статьи](#)
- [plugin_development_icon](#)

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



Permanent link:
https://wwoss.ru/doku.php?id=wiki:devel:syntax_plugin_skeleton

Last update: **2025/01/09 13:05**

