

Action Plugins

[Action Plugins](#) предназначены, чтобы работать с [событиями](#) Dokuwiki и иметь возможность добавить свой расширенный функционал в любой части [Dokuwiki](#).

Описание

Action plugins загружаются до того, как вики делает что-либо значительное. Сразу же после загрузки, вики вызывает метод `register()` каждого плагина для того чтобы дать ему возможность зарегистрировать свои обработчики событий. Когда происходит некоторое событие, все зарегистрированные на него обработчики вызываются по очереди (в произвольном порядке) и им передается по ссылке объект события. Обработчик получает возможность выполнить какое-либо действие на основании данных, хранящихся в этом объекте, модифицировать эти данные или повлиять на дальнейшую обработку события. На странице [events](#) можно узнать больше о том, как устроена система событий, а также посмотреть их полный список.

Техническое описание

Action plugins используют такой же основной формат и правила наименования, как и другие типы плагинов для Dokuwiki.

- каждый плагин должен располагаться в собственной директории внутри каталога `lib/plugins`
- скрипт плагина должен называться `action.php`, или должен располагаться в подкаталоге `action`, и тогда называться как угодно, смотрите [файловую структуру](#).
- плагин должен содержать один класс `action_plugin_<plugin name>` как расширение базового класса `DokuWiki_Action_Plugin`, который можно найти в `lib/plugins/action.php`. Если файл плагина располагается в подкаталоге `action`, тогда класс необходимо назвать `action_plugin_<plugin name>_<action plugin filename>` (где `action_plugin_filename` не должно содержать расширение `'.php'`).
- плагин поддерживает стандартные introspection, локализации и конфигурации через базовый класс `DokuWiki_Plugin`, который располагается в файле `inc/plugin.php`, смотрите [Встроенные функции плагина](#).
- плагин должен иметь два метода, `getInfo()` и `register()`.
- внешние библиотеки могут быть загружены в тот момент, когда это необходимо плагину или в конструкторе, но не в самом начале файла

getInfo()

!!! Теперь вместо неё прилагается отдельный файл `plugin.info.txt` !!!

обязательный

```
function getInfo(){
    return array(
        'author' => '<имя автора плагина>',
        'email'  => '<контактный e-mail автора плагина>',
        'date'   => '<дата версии плагина>',
        'name'   => '<название плагина>',
        'desc'   => '<описание плагина (поддерживается многострочие)',
        'url'    => '<домашняя страничка плагина: ссылка, где можно найти
информацию по плагину>',
    );
}
```

register()

обязательный

```
/**
 * plugin should use this method to register its handlers with the
 * dokuwiki's event controller
 *
 * @param $controller Dokuwiki's event controller object. Also
 * available as global $EVENT_HANDLER
 *
 * @return not required
 */
function register(Doku_Event_Handler $controller) {
    $controller->register_hook(<EVENT NAME>, <EVENT ADVISE>, $this, <event
handler function>, <parameters to be passed to event handler>);
}
```

<event handler>()

необязательный их может быть любое количество; может быть назван любым именем, которое не используется этим плагином или его предками

```
/**
 * custom event handler
 *
 * @param $param (mixed) the parameters passed to register_hook
 * when this handler was registered
 * @param $event (object) event object by reference
 *
 * @return not required
 */
function <event_handler>(&$event, $param) {
    // custom script statements ...
}
```

Examples

Action plugins are a way to modify many aspects of how DokuWiki behaves in certain cases independent of a page's syntax. To be able to modify a DokuWiki internal behavior it needs to trigger an event. Your action plugin can register as a handler for such an event and then work with the given event data.

To learn more about events, read the following pages:

- [events](#)
- [events list](#)
- [event handlers](#)

Sample action plugin 1

Insert a javascript script link in all pages.

- Register the [TPL_METAHEADER_OUTPUT](#) event, with a before EVENT_ADVICE.
- Add javascript information to «script» meta headers as array type.

[action.php](#)

```
<?php
/**
 * Example Action Plugin:   Example Component.
 *
 * @author      Samuele Tognini <samuele@cli.di.unipi.it>
 */

if(!defined('DOKU_INC')) die();
if(!defined('DOKU_PLUGIN'))
define('DOKU_PLUGIN',DOKU_INC.'lib/plugins/');
require_once(DOKU_PLUGIN.'action.php');

class action_plugin_example extends DokuWiki_Action_Plugin {

    /**
     * return some info
     */
    function getInfo(){
        return array(
            'author' => 'Me name',
            'email'  => 'myname@example.org',
            'date'   => '2006-12-17',
            'name'   => 'Example (action plugin component)',
            'desc'   => 'Example action functions.',
            'url'    => 'http://www.example.org',
        );
    }
}
```

```

/**
 * Register its handlers with the dokuwiki's event controller
 */
function register(Doku_Event_Handler $controller) {
    $controller->register_hook('TPL_METAHEADER_OUTPUT', 'BEFORE',
$this, '_hookjs');
}

/**
 * Hook js script into page headers.
 *
 * @author Samuele Tognini <samuele@cli.di.unipi.it>
 */
function _hookjs(&$event, $param) {
    $event->data["script"][] = array ("type" => "text/javascript",
                                     "charset" => "utf-8",
                                     "_data" => "",
                                     "src" =>
DOKU_BASE."lib/plugins/example/example.js"
    );
}
}

```

Sample Action Plugin 2

Inserts a button into the editor toolbar:

- registers as handler for the `TOOLBAR_DEFINE` event with an AFTER advise
- adds a button definition to the event's data

[action.php](#)

```

<?php
/**
 * Example Action Plugin: Inserts a button into the toolbar
 *
 * @author Gina Haeussge <osd@foosel.net>
 */

if (!defined('DOKU_INC')) die();
if (!defined('DOKU_PLUGIN')) define('DOKU_PLUGIN', DOKU_INC .
'lib/plugins/');
require_once (DOKU_PLUGIN . 'action.php');

class action_plugin_actionexample extends DokuWiki_Action_Plugin {

    /**

```

```
* Return some info
*/
function getInfo() {
    return array (
        'author' => 'Some name',
        'email' => 'foo@bar.org',
        'date' => '2007-04-05',
        'name' => 'Toolbar Action Plugin',
        'desc' => 'Inserts a button into the toolbar',
        'url' => 'http://www.example.com/plugin/toolbar',
    );
}

/**
* Register the eventhandlers
*/
function register(Doku_Event_Handler $controller) {
    $controller->register_hook('TOOLBAR_DEFINE', 'AFTER', $this,
    'insert_button', array ());
}

/**
* Inserts the toolbar button
*/
function insert_button(& $event, $param) {
    global $lang;
    global $conf;
    include_once (dirname(__FILE__) . '/lang/en/lang.php');
    @ include_once (dirname(__FILE__) . '/lang/' . $conf['lang'] .
    '/lang.php');

    $event->data[] = array (
        'type' => 'format',
        'title' => $lang['qb_abutton'],
        'icon' => '../..../plugins/actionexample/abutton.png',
        'open' => '<abutton>',
        'close' => '</abutton>',
    );
}
}
```

https://www.dokuwiki.org/wiki:devel:action_plugins

From:

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

Permanent link:

https://wwoss.ru/doku.php?id=wiki:devel:action_plugins

Last update: **2024/08/15 01:53**

