

# The Event Handler

action.php

```
<?php

/**
 * Действие по добавлению кнопки DW Edit в меню страницы
 * Позволяет напрямую переключаться в родной редактор «Докувики».
 * без предварительного выхода из ckeditor
 * @author Майрон Тернер <mturnerATmtturnerDOTorg>
 */

if (!defined('DOKU_INC'))
{
    die();
}

class action_plugin_dwedit extends DokuWiki_Action_Plugin
{
    var $ckgedit_loaded = false;
    var $helper;

    function __construct() {
        /* is either ckgdoku or ckgedit enabled,
        if so get a reference to the helper */
        $list = plugin_list('helper');
        if(in_array('ckgedit',$list)) {
            $this->ckgedit_loaded=true;
            $this->helper = plugin_load('helper', 'ckgedit');
        }
        else if(in_array('ckgdoku',$list)) {
            $this->ckgedit_loaded=true;
            $this->helper = plugin_load('helper', 'ckgdoku');
        }
    }

    function register(Doku_Event_Handler $controller)
    {
        $controller->register_hook('MENU_ITEMS_ASSEMBLY', 'AFTER',
            $this, 'addsvgbutton', array());
        /* discontinued/deprecated hooks */
        $controller->register_hook('TEMPLATE_PAGETOOLS_DISPLAY', 'BEFORE',
            $this, 'dwedit_action_link', array('page_tools'));
        $controller->register_hook('TEMPLATE_DWEDITLINK_DISPLAY', 'BEFORE',
            $this, 'dwedit_action_link', array('user'));
    }
}
```

```

public function addsvgbutton(Doku_Event $event) {
    /* если это не страница ИЛИ ckgedit/ckgedoku не активна -> return */
    if($event->data['view'] != 'page' || !$this->ckgedit_loaded) return;

    // получить имя кнопки из включенного в данный момент плагина ckg_* plugin
    $btn = $this->helper->getLang('btn_dw_edit');
    if(!$btn) $btn = 'DW Edit';
    array_splice($event->data['items'], -1, 0,
        [new \dokuwiki\plugin\dwedit\MenuItem($btn)]);
}

function dwedit_action_link(&$event, $param)
{
    /*
    код идет сюда для протокола предыдущего пункта меню
    */
}
}

```

## The Menu Item

### Menuitem.php

```

<?php

namespace dokuwiki\plugin\dwedit;

use dokuwiki\Menu\Item\AbstractItem;

/**
 * Class MenuItem
 *
 * @package dokuwiki\plugin\dwedit
 */
class MenuItem extends AbstractItem {

    /** @var string do action for this plugin */
    protected $type = '';
    private $btn_name;

    /** @var string icon file */
    protected $svg = __DIR__ . '/edit_pencil.svg';

    /**
     * MenuItem constructor.
     * @param string $btn_name (can be passed in from the event
     handler)
     */
}

```

```

public function __construct($btn_name = "") {
    parent::__construct();
    global $REV, $INFO;

    if($btn_name) {
        $this->btn_name = $btn_name;
    }

    if($REV) $this->params['rev'] = $REV;

    /*switching over to the native dw editor requires
       two additional http paramters */
    $this->params['mode'] = 'dwiki';
    $this->params['fck_preview_mode'] = 'nil';

    // use alternate icon if user does not have edit permission
    if ($INFO['perm'] < AUTH_EDIT) {
        $this->svg = __DIR__ . '/dwedit_view.svg';
    }
}

/**
 * Get label from plugin language file
 *
 * @return string
 */
public function getLabel() {
    if($this->btn_name) return $this->btn_name;
}

/*
    if the button name has not been set up in the constructor
    you can get it now.
*/
    $hlp = plugin_load('action', 'dwedit');
    return $hlp->getLang('btn_dw_edit');
}
}

```

## Attaching a Javascript Url to link or button

```

<?php

namespace dokuwiki\plugin\overlay;

use dokuwiki\Menu\Item\AbstractItem;

/**

```

```
* Class MenuItem
*
* Implements the overlay button for DokuWiki's menu system
*
* @package dokuwiki\plugin\overlay
*/
class MenuItem extends AbstractItem {

    /** @var string do action for this plugin */
    protected $type = '';
    private $btn_name;

    /** @var string icon file */
    protected $svg = __DIR__ . '/screen-frame.svg';
    /**
     * MenuItem constructor.
     * @param string $btn_name (can be passed in from the event handler)
     */
    public function __construct($btn_name = "") {
        parent::__construct();
        $this->params['do']="";
        if($btn_name) {
            $this->btn_name = $btn_name;
        }
    }

    /**
     * Get label from plugin language file
     * @return string
     */
    public function getLabel() {
        if($this->btn_name) return $this->btn_name;
        $hlp = plugin_load('action', 'overlay');
        return $hlp->getLang('btn_dw_edit');
    }

    public function getLink() {
        return 'javascript:jQuery("#overlay").toggle();void(0)';
    }
}
```

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

Permanent link:  
<https://wwoss.ru/doku.php?id=wiki:devel:menus:example>

Last update: **2023/08/17 13:11**

