



Dilbert Daily Cartoon plugin

I try to include the Dilbert Daily Cartoon on a  [DokuWiki](#) page. Very simple, let's take SimplePie (included with  [DokuWiki](#)), the URL and on we go.

Very funny. On the [demo page of SimplePie](#) everything looks good. But at home? Nothing, "no feed found". So, let's download the actual version (which includes this demo page), install it locally and – "no feed found". After some searching on [SimplePie's web site](#) I found out that Feedburner does not like SimplePie, so that I have to fake the user agent string...

OK, let's go on. After some hints from Andi Gohr via the mailing list I managed to get it running with the following code:

```
<?php
/**
 * dilbert plugin: shows the daily cartoon from dilbert.com
 */
/**
 * @license    GPL 2 (http://www.gnu.org/licenses/gpl.html)
 * @author     Werner Flamme <w.flamme@web.de>
 * @date       2009-02-02
 */

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

/**
 * All DokuWiki plugins to extend the parser/rendering mechanism
 * need to inherit from this class
 */
class syntax_plugin_dilbert extends DokuWiki_Syntax_Plugin {

    /**
     * return some info
     * @return array (hash) with some info about the plugin
     */
    function getInfo()
    {
        return array(
            'author' => 'Werner Flamme',
            'email'  => 'w.flamme@web.de',
            'date'   => '2009-02-02',
            'name'   => 'Dilbert Daily Cartoon Plugin',
            'desc'   => 'Shows the daily cartoon from dilbert.com ' .
                        'as shown on
http://feedproxy.google.com/~r/DilbertDailyStrip/<...>',

```

```
        'url'      =>
'http://www.wernerflamme.name/doku.php?id=comp:dilbert'
    );
} // function getInfo

/**
 * What kind of syntax are we?
 * @return string containing the syntax type
 */
function getType()
{
    return 'substitution';
} // function getType

/**
 * What kind of plugin are we?
 * @return string containing the kind of the plugin
 */
function getPType()
{
    return 'block';
} // function getPType

/**
 * Where to sort in?
 * @return integer number giving the sort sequence number
 */
function getSort()
{
    return 200;
} // function getSort

/**
 * Connect pattern to lexer
 */
function connectTo($mode)
{
    $this->Lexer->addSpecialPattern('{~dilbert~}', $mode,
'plugin_dilbert');
} // function connectTo

/**
 * Handle the match
 */
function handle($match, $state, $pos)
{
    return array();
} // function handle

/**
 * Create output
```

```

* @param $mode      current mode of DokuWiki
*
* (see http://wiki.splitbrain.org/plugin:tutorial)
* @param $renderer  DokuWiki's rendering object
* @param $data      (not looked at)
* @return true, if rendering happens, false in all other cases
**/
function render($mode, &$renderer, $data)
{
    if ($mode == 'xhtml') {
        // we need the SimplePie library
        require_once(DOKU_INC . 'inc/FeedParser.php');
        // where to look for the feed:
        //$url =
'http://feeds.feedburner.com/DilbertDailyStrip?format=xml';
        $url = 'http://feedproxy.google.com/DilbertDailyStrip';
        // create SimplePie feed parsing object
        $feed = new FeedParser();
        // next line is mandatory, since feedburner dislikes simplepie
        $feed->set_useragent('Mozilla/4.5 (as DokuWiki plugin)');
        // point feed to URL
        $feed->set_feed_url($url);
        // get data
        $feed->init();
        // ...and mangle^Wmanage it :-)
        $feed->handle_content_type();
        // we only want the cartoon that was published during last 24
hours
        $yesterday = time() - (24 * 60 * 60);
        // loop at the items in the feed
        foreach ($feed->get_items() as $item) {
            // if the item has been published during the last 24
hours...
            if ($item->get_date('U') > $yesterday) {
                $feedDescription = hsc($item->get_description());
                $image
                =
$this->_returnImage($feedDescription);
                $imageurl
                = $this->_scrapeImage($image);
                $src
                = $imageurl;
                $title
                = 'Dilbert Daily Cartoon';
                $align
                = null;
                $width
                = null;
                $height
                = null;
                $cache
                = false;
                $renderer->externalmedia($src, $title, $align, $width,
$height, $cache);
            } // if ($item->get_date('U') > $yesterday)
        } // foreach ($feed->get_items() as $item)
        return true;
    } // if ($mode == 'xhtml')
    return false;
} // function render

```

```

/**
 * taken from esteban on
 http://simplepie.org/support/viewtopic.php?id=643
 * Last edited by esteban (23 April 2007 03:06:30)
 *
 * Get an image
 * @return (string) content of image tag in the feed
 */
function _returnImage($text)
{
    $text = html_entity_decode($text, ENT_QUOTES, 'UTF-8');
    $pattern = "<img[^>]+\>/i";
    preg_match($pattern, $text, $matches);
    $text = $matches[0];
    return urldecode($text);
} // function _returnImage

/**
 * taken from esteban on
 http://simplepie.org/support/viewtopic.php?id=643
 * Last edited by esteban (23 April 2007 03:06:30)
 *
 * Filter out image url only
 * @return (string) URL of the picture
 */
function _scrapeImage($text)
{
    $pattern = '/src=([\'"])?([^\\">]+)([\'"]>)/';
    preg_match($pattern, $text, $link);
    $link = $link[1];
    return hsc($link);
} // function _scrapeImage

} // class syntax_plugin_dilbert

//Setup VIM: ex: et ts=4 enc=utf-8 :

```

The function `hsc()` is a wrapper around `htmlspecialchars()` defined in `inc/common.php`.

From:
<http://wernerflamme.net/> - **Werners Wiki**

Permanent link:
<http://wernerflamme.net/doku.php?id=comp:en:dilbert>

Last update: **2009-06-22 2039**

