

Writing a PHP Plugin Widget for WordPress

Some useful resources:

<http://blue-anvil.com/archives/create-a-wordpress-recent-posts-widget>

<http://lonewolf-online.net/computers/knowledgebase/wordpress-how-to-create-widgets/>

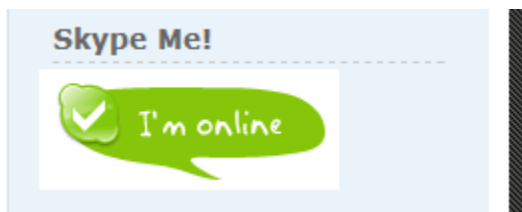
<http://guff.szub.net/2006/04/06/my-widget-example-wordpress-widget/>

Aim of my plugin:

With this plugin I intent to create a “Skype Button” that will be customisable to the user:

<http://www.skype.com/intl/en-gb/share/buttons/>

The user will install and activate the plugin, input their Skype username and chose the favoured button type, they will also be able to customise the title and text underneath the button. This will then display their Skype Button in their sidebar, like this:



Writing the plugin:

This is the code you need to start your plugin:

```
<?php
/*
Plugin Name: Skype Status Button
Plugin URI: http://315cmc.tomupton.com/?page_id=68
Description: Skype Status Button Widget Plugin - shows your online status for Skype
Author: Tom Upton
Version: 1.0
Author URI: http://315cmc.tomupton.com/
*/
```

The first part, “<?php”, starts the PHP. The next part is what is displayed in the plugin control panel in WordPress, this codes shows this in WordPress:

Skype Status Button	1.0	Skype Status Button Widget Plugin – shows your online status for Skype <i>By Tom Upton.</i>	Deactivate	Edit
----------------------------	-----	---	------------	------

I think the parts in orange are fairly self explanatory, so I will not expand on this.

This code checks that the sidebar widget functions required are available, if not the code will not execute any further:

Writing a plugin widget for WordPress using PHP

```
function widget_SkypeButton_init() {  
  
    if ( !function_exists('register_sidebar_widget') || !function_exists('register_widget_control') )  
        return;
```

This next lot of code creates the sidebar widget and helps it conform to the active theme:

```
function widget_SkypeButton($args) {  
  
    extract($args);
```

This code collects the widgets options and defines the defaults:

```
$options = get_option('widget_SkypeButton');  
$title = empty($options['title']) ? 'My Skype Button:' : $options['title'];  
$user = empty($options['user']) ? 'Username' : $options['user'];  
$button = empty($options['button']) ? 'balloon' : $options['button'];  
$text = empty($options['text']) ? '' : $options['text'];  
$width = empty($options['width']) ? '' : $options['width'];  
$height = empty($options['height']) ? '' : $options['height'];
```

The first option is the title, then your Skype username, balloon type, user defined text, and the width and height of the button (as this changes depending on the type of balloon chosen).

The next piece of code is what is shown by the widget:

```
echo $before_widget;  
echo $before_title . $title . $after_title;  
echo '<script type="text/javascript"  
src="http://download.skype.com/share/skypebuttons/js/skypeCheck.js"></script><a  
href="skype:'. $user. '?call"></a><br />';  
echo $text;  
echo $after_widget;  
}
```

In this case it puts the title first, followed by the Skype button, with the user defined options like username and balloon type, then the text and then it closes the widget.

The user can define variables such as their username and desired text in the options box, which is what the next part of code creates:

```
function widget_SkypeButton_control() {  
  
    $options = $newoptions = get_option('widget_SkypeButton');  
  
    if ( $_POST['SkypeButton-submit'] ) {  
        $newoptions['title'] = strip_tags(strip_slashes($_POST['SkypeButton-title']));  
        $newoptions['user'] = strip_tags(strip_slashes($_POST['SkypeButton-user']));  
        $newoptions['button'] = strip_tags(strip_slashes($_POST['SkypeButton-button']));  
        $newoptions['text'] = strip_tags(strip_slashes($_POST['SkypeButton-text']));  
        $newoptions['width'] = strip_tags(strip_slashes($_POST['SkypeButton-width']));
```

Writing a plugin widget for WordPress using PHP

```
$newoptions['hight'] = strip_tags(stripslashes($_POST['SkypeButton-hight']));
}

if ( $options != $newoptions ) {
    $options = $newoptions;
    update_option('widget_SkypeButton', $options);
}
```

This code formats the options as valid HTML, to reduce the chance of it causing problems:

```
$title = htmlspecialchars($options['title'], ENT_QUOTES);
$user = htmlspecialchars($options['user'], ENT_QUOTES);
$button = htmlspecialchars($options['button'], ENT_QUOTES);
$text = htmlspecialchars($options['text'], ENT_QUOTES);
$width = htmlspecialchars($options['width'], ENT_QUOTES);
$hight = htmlspecialchars($options['hight'], ENT_QUOTES);
```

The code required to create the options box that is seen within the widgets panel is shown next:

```
?>
    <div>
    <label for="SkypeButton-title" style="line-height:35px;display:block;">Widget title:
    <input name="SkypeButton-title" type="text" id="SkypeButton-title" value="<?php echo
$title; ?>" size="40" />
    </label>
    <label for="SkypeButton-user" style="line-height:35px;display:block;">Your Skype
Username:
    <input name="SkypeButton-user" size="40" id="SkypeButton-user" value="<?php echo
$user; ?>" />
    <option value="balloon"
<?=( $button=="balloon" )?'selected':"?"><?=( $width=="150" )?'selected':"?"><?=( $hight=="60" )?'selected':"?">>B
alloon</option>
    <option value="bigclassic"
<?=( $button=="bigclassic" )?'selected':"?"><?=( $width=="182" )?'selected':"?"><?=( $hight=="44" )?'selected':"?">>
Big Classic</option>
    </select>
    </label>
    <label for="SkypeButton-text" style="line-height:35px;display:block;">Your Skype Username:
    <textarea name="SkypeButton-text" cols="37" rows="7" id="SkypeButton-text"><?php echo
$text; ?></textarea>
    </label>
    <input type="hidden" name="SkypeButton-submit" id="SkypeButton-submit" value="1" />
    </div>
<?php
}
```

The final code here does several things; it registers the widget and the options box as well as delaying the plugin execution until the sidebar has loaded first, also closes the PHP tags opened at the start:

```
register_sidebar_widget('Skype Button', 'widget_SkypeButton');

register_widget_control('Skype Button', 'widget_SkypeButton_control');
```

Writing a plugin widget for WordPress using PHP

```
}  
  
add_action('plugins_loaded', 'widget_SkypeButton_init');  
?>
```

With reasonable knowledge of HTML and CSS, and no prior familiarity with PHP, I have managed to write this plugin widget for WordPress, that displays a users Skype status in their sidebar, giving them the option of two kinds of “balloon”, a personalised title and text box.

This plugin has been fully tested using WordPress 2.3.3 with Firefox 2.0.0.14 and it works fine. I have trialled it in WordPress 2.5, it works but requires some tweaking due to the different plugin admin page.