

Desarrollo De Widget En WordPress Para Contador De Redes Sociales

Anteriormente habíamos hablado de como [obtener el numero de los seguidores en Facebook, Twitter y G+](#) . Para esto, se ha creado **tres funciones de PHP** para simplificar la recuperación de estos números de las 3 **redes sociales mas usadas**.

Estas funciones aceptan como argumento, el nombre de una página de Facebook, nombre de usuario de Twitter y Google +, o el ID respectivamente, con estos datos las funciones retornan los números de los seguidores que hay en cada perfil social. Estas son funciones bastantes sencillas que te ayudan a **obtener el numero de seguidores** y así construir tu propio widget de forma fácil.

En este articulo complementaremos el anterior, ya que vamos a **desarrollar un widget de redes sociales** que se enlace con una página en Facebook, Twitter y Google+.

El resultado final de este tutorial será el siguiente:



Codificando El

Widget

Al igual que con todos los **plugins de WordPress**, el encabezado del archivo contiene meta información (como el nombre, versión, autor, etc.) en relación con el plugin debe estar al principio del archivo plugin.

A continuación se muestra la información importante que debe tener este plugin, puedes modificarlo a tu gusto:

```
[php]

<?php
/*
 * Plugin Name: Nombre del plugin
 * Plugin URI: tu URL
 * Description: Aguna descripcion
 * Author: nombre del autor
 * Author URI: URL del autor
 */

[/php]
```

Para crear un widget de WordPress, necesitas heredar de la clase WP_Widget y usar las funciones necesarias en la clase hija. Por último, la clase hija del widget se habrá registrado y conectado a WordPress.

```
[php]

class <strong>nombreDeLaClase</strong> extends WP_Widget {
// ...

[/php]
```

Dé al widget un nombre y una descripción mediante la **función __construct ()**.

```
[php]
```

```
function __construct() {  
  
parent::__construct(  
'<strong>nombreDeLaClase</strong>', __('nombreDelWidget',  
'translation_domain' ),  
array( 'description' => __('La descripcion que deseas',  
'translation_domain' ) )  
);  
  
}
```

```
[/php]
```

En este punto, es el momento de tomar ventaja de las funciones de ayuda que se han creado en el [anterior articulo](#) para que podamos mostrar los likes y Followers de las respectivas redes sociales.

```
[php]
```

```
public function twitter_count( $username ) {  
  
require_once 'TwitterAPIExchange.php';  
  
// Set access tokens here – see: https://dev.twitter.com/apps/  
$settings = array(  
    'oauth_access_token'           => "211978035-  
fedllb5xEQhnoHxAsK3259VI0hsFrLuRUyR4Atvr",  
    'oauth_access_token_secret'    => "  
7Nev2Ey0xoHAVgb8Y5VHPRYuKbKomFqe3kf1ou0HtmHVs",  
    'consumer_key' => "MsHrMc5B9dZyP8mgqV0m2JGsq",  
    'consumer_secret'              => "  
YhIdWozaAb9cvKcjKqamEcN2GgSBrzqfWZpIvKSeYVvCQsb8LL"  
);  
  
$url = 'https://api.twitter.com/1.1/users/show.json';  
$getfield = '?screen_name=' . $username;  
$request_method = 'GET';
```

```

$twitter_instance = new TwitterAPIExchange( $settings );
$follow_count = $twitter_instance ->setGetfield( $getfield )
->buildOauth( $url, $request_method )
->performRequest();
$count = json_decode( $follow_count, true );

return $count['followers_count'];

}

```

[/php]

[php]

```

public function facebook_count( $username ) {
$facebook_count          =          file_get_contents(
'http://graph.facebook.com/' . $username );
return json_decode( $facebook_count )->likes;
}

```

[/php]

[php]

```

public function googleplus_count( $username, $apikey =
'AIzaSyBHm7J9qLupabYWaxLg_9_UZPbxWdso2vY' ) {
$google                  =          file_get_contents(
'https://www.googleapis.com/plus/v1/people/' . $username .
'?key=' . $apikey );
return json_decode( $google ) -> circledByCount;
}

```

[/php]

En la funcion twitter_count(), el token del acceso a la aplicación **OAuth de Twitter**, el modo de acceso secreto, la clave de consumidores y la clave secreta de consumidores son

todos usados en la función. Tendrás que cambiar a las **credenciales de OAuth de su aplicación de Twitter**. Lo mismo va para la clave de la **API de Google Plus** en la función `googleplus_count()`.

Las opciones de configuración de widgets consistirán en cinco campos de formulario que contienen el título de los nombres de usuario de Facebook, Twitter, Google+, tal como se muestra a continuación:



El siguiente código es de la **función form()**, la cual es la responsable de que se muestre lo anterior:

[php]

```
public function form( $instance ) {
isset ( $instance['title'] ) ? $title = $instance['title'] :
null;
empty ( $instance['title'] ) ? $title = 'My Social Profile' :
null;

isset ( $instance['facebook'] ) ? $facebook =
$instance['facebook'] : null;
isset ( $instance['twitter'] ) ? $twitter =
$instance['twitter'] : null;
isset ( $instance['google'] ) ? $google = $instance['google']
: null;
?>
<p>
<label for="<?php echo $this->get_field_id('title'); ?>"><?php
_e('Title:'); ?></label>
<input class="widefat" id="<?php echo $this->get_field_id(
'title' ); ?>" name="<?php echo $this->get_field_name( 'title'
); ?>" type="text" value="<?php echo esc_attr( $title ); ?>">
</p>

<p>
```

```

<label for="<?php echo $this->get_field_id('facebook');
?>"><?php _e('Facebook Page Username:'); ?></label>
<input class="widefat" id="<?php echo
$this->get_field_id('facebook'); ?>" name="<?php echo
$this->get_field_name('facebook'); ?>" type="text"
value="<?php echo esc_attr( $facebook ); ?>">
</p>

```

```

<p>
<label for="<?php echo $this->get_field_id('twitter');
?>"><?php _e('Twitter Username:'); ?></label>
<input class="widefat" id="<?php echo
$this->get_field_id('twitter'); ?>" name="<?php echo
$this->get_field_name('twitter'); ?>" type="text" value="<?php
echo esc_attr( $twitter ); ?>">
</p>

```

```

<p>
<label for="<?php echo $this->get_field_id('google');
?>"><?php _e('Google+ Username or ID:'); ?></label>
<input class="widefat" id="<?php echo $this->get_field_id(
'google' ); ?>" name="<?php echo $this->get_field_name(
'google' ); ?>" type="text" value="<?php echo esc_attr(
$google ); ?>">
</p>

```

```

<?php
}

```

```
[/php]
```

Cuando los valores se introducen en el campo de formulario, es necesario que se guarde en la base de datos. La **función update()** valida los valores del formulario **excluyendo datos maliciosos** y guarda los valores desinfectados a la base de datos.

```
[php]
```

```

public function update( $new_instance, $old_instance ) {

$instance = array();
$instance['title'] = ( ! empty ( $new_instance['title'] ) ) ?
strip_tags( $new_instance['title'] ) : »;
$instance['facebook'] = ( ! empty ( $new_instance['facebook'] ) ) ?
strip_tags( $new_instance['facebook'] ) : »;
$instance['twitter'] = ( ! empty ( $new_instance['twitter'] ) )
? strip_tags( $new_instance['twitter'] ) : »;
$instance['google'] = ( ! empty ( $new_instance['google'] ) ) ?
strip_tags( $new_instance['google'] ) : »;
$instance['linkedin'] = ( ! empty ( $new_instance['linkedin'] ) )
? strip_tags( $new_instance['linkedin'] ) : »;

return $instance;

}

[/php]

```

El siguiente es la función widget() que muestra los enlaces a perfiles de redes sociales, junto con los likes y followers.

```

[php]

$title = apply_filters( 'widget_title', $instance['title'] );
$facebook = $instance['facebook'];
$twitter = $instance['twitter'];
$google = $instance['google'];

// social profile link
$social_widget = '
<ul class="diverz">
<li class="diverz facebookz">
<a href="https://www.facebook.com/' . $facebook . '">
<div class="main-diverz">
<i class="facebookz fa fa-facebook"></i>
<br/>
<big class="spanz facebookz">facebook</big>

```

```

</div>
</a>
<div class="sub-diverz">
<strong>' .
$this->facebook_count( $facebook ) . '</strong>
<br/>fans
</div>
</li>

<li class="diverz twitterz">
<a href="https://www.twitter.com/' . $twitter . '">
<div class="main-diverz">
<i class="twitterz fa fa-twitter"></i><br/>
<big class="spanz twitterz">twitter</big>
</div></a>
<div class="sub-diverz"><strong>' .
$this->twitter_count('tech4sky') . '</strong><br/>fans
</div>
</li>

<li class="diverz googlez">
<a href="https://plus.google.com/u/0/' . $google . '">
<div class="main-diverz">
<i class="googlez fa fa-google-plus"></i><br/>
<big class="spanz googlez">google+</big>
</div></a>
<div class="sub-diverz"><strong>' .
$this->googleplus_count( $google ) . '</strong><br/>fans
</div>
</li>
</ul>';

echo $args['before_widget'];

if ( !empty($title) ) {
echo $args['before_title'] . $title . $args['after_title'];
}

```



```
echo $social_widget;  
  
echo $args['after_widget'];  
}
```

```
[/php]
```

Hemos terminado incluyendo los métodos y códigos necesarios a la clase, todo lo que queda es incluir la llave de cierre de clase.

```
[php]
```

```
// ...  
}
```

```
[/php]
```

Ya tenemos el código necesario para que funcione en nuestro blog, ahora necesitamos registrar el nombre de la clase del widget usando la **función register_widget de WordPress**, y luego añadir a widgets_init.

Si no haces esto, el widget no será reconocido por WordPress.

```
[php]
```

```
function register_tutsplus_social_profile() {  
    register_widget( 'Tutsplus_Social_Profile' );  
}  
add_action( 'widgets_init', 'register_tutsplus_social_profile'  
);
```

```
[/php]
```

Para dar el widget un estilo agradable, a continuación puedes encontrar el CSS del widget, que también incluye el archivo de hoja de estilo iconos [importada de Font Awesome](#), si tienes conocimientos básicos de CSS puedes personalizar aún más el widget.

```
[css]
```

```
@import                                "//netdna.bootstrapcdn.com/font-
awesome/4.1.0/css/font-awesome.min.css";

social-icons {
font-size: 21px;
}

ul .diverz {
width: 80px;
border: 1px solid #e3e3e3;
list-style-type: none;
overflow: hidden;
padding: 4px 2px;
margin: 2px 2px !important;
background-color: #eee;
}
.main-diverz {

font-size: 16px;
padding: 2px;
}

.sub-diverz {

color: #000;
background-color: #FFFFFF;
margin: 2px;
padding: 2px;
}

.diverz {
float: left;
text-align: center;
}

.spanz {
font-size: 15px;
vertical-align: middle;
}
```

```
.facebookz, .facebookz a {  
color: #3B5998;  
}
```

```
.twitterz, .twitterz a {  
color: #00abe3;  
}
```

```
.main-diverz i {  
  
font-size: 20px;  
}
```

```
[/css]
```

Guarde el archivo CSS con el nombre que desees, este debe estar en la carpeta raíz del plugin. Por último, tenemos que poner en cola el archivo CSS para WordPress:

```
[php]
```

```
// enqueue css stylesheet  
function nombreDeLaFuncionCSS() {  
wp_enqueue_style( 'nombreDelWidget', plugins_url(  
'nombreArchivoCSS.css', __FILE__ ) );  
}  
add_action( 'wp_enqueue_scripts', 'nombreDeLaFuncionCSS' );
```

```
[/php]
```

Listo! Hemos terminado.

Finalmente

Como una sugerencia, puedes extender los beneficios de este widget a otras redes sociales, y mostrar cuan importante es tu sitio. Puedes personalizar el estilo del Widget a tu gusto, y darle una mejor y llamativa apariencia.