aw2.shortcode

Syntax

```
[aw2.shortcode  shortcode_name]

    // Shortcode logic goes here
    // Shortcode parameters are accessible using in sc.prameter_name syntax.

[/aw2.shortcode]
```

Description

Shortcodes are “WordPress-specific code” which provides an easy way to insert complex logic into posts, pages, modules e.t.c. The logic it contains may be right from displaying a simple HTML content, to fully functional e-commerce store or even an audio-video playlist that end user interacts with.

To give an e.g. in Awesome Studio there is a built-in shortcode called [notification.send] . The notification sending logic is bundled into that shortcode and can be used in the page, post or even in an awesome studio module, which basically sends notification through an e-mail or SMS, to the intended recipient.

Following is the syntax of [notification.send] shortcode,

[notification.send to=’to_email_address’ from=’from_email_address’ // some more parameters /]

As we can see this shortcode has “to” and “from” parameters, which basically are set to send the notification ‘to’ the intended recipient “from” the sender.

The [aw2.shortcode] is a wrapper “shortcode” provided by Awesome Studio platform to register custom shortcodes along with its handler. You can use this, instead of the default WordPress way of registering shortcodes and its handlers.

As we see from the syntax above, the only parameter to[aw2.shortcode] is the name of the shortcode which is then to be used later, for executing that particular shortcode.

The parameter which is sent from shortcode call, are accessed inside the shortcode logic using sc.parameter_name syntax. In our example of notification.send shortcode to the parameter set from shortcode call can be accessed using sc.to for “to” parameter and sc.from for “from” parameter.

Parameters

#shortcode_name

(string) (Required) The name of the shortcode to be created.

Usage

Following code is creates a shortcode called [notification.send] (which is inbuilt shortcode of Awesome Studio), shortcode definition and shortcode usage are discussed.

Shortcode definition

```
[aw2.shortcode notification.send]
 [aw2.query get_post post_slug='{sc.mail_slug}' post_type=as_notify set='sc.mail' /]

// If empty sc.from e-mail address use admin_email address
 [aw2.empty sc.from]
   [aw2.set sc.from="{{aw2.get function.get_option p1='blogname'}} <{{aw2.get function.get_option p1='admin_email'}}>" /]
 [/aw2.empty]

/* If not empty sc.to e-mail address, set to, cc, from, subject e-mail parameters passed in to aw2.wp_mail shortcode. */

/* Also check if if there is any attachment to the e-mail which is to be sent. */

 [aw2.not_empty sc.to]
   [aw2.wp_mail part=start]
   {
    "to":"[aw2.get sc.to /]",
    "cc":"[aw2.get sc.cc /]",
    "from": "[aw2.get sc.from /]",
    "subject":"[aw2.get sc.mail.meta.subject.run /]"
    [aw2.if not_empty="{{aw2.get sc.attachment}}" ],
    "attachment":"[aw2.get sc.attachment /]"
    [/aw2.if]
   }
   [/aw2.wp_mail]

   //Set the body for the e-mail.
   [aw2.wp_mail part=message]
     [aw2.get sc.mail.parse_content /]
   [/aw2.wp_mail]

  //Send the mail using wp_mail.
   [aw2.wp_mail part=run][/aw2.wp_mail]
 [/aw2.not_empty]
 [aw2.get sc.mail.meta.sms.run set='sc.sms' /]

 // Check if site setting for sending SMS is set to on and if mobile
 [aw2.if cond="{{aw2.get site_settings.send_sms}}" equal="on"]
 [/aw2.if]
 [aw2.and not_empty="{{aw2.get sc.mobile}}"]
 [/aw2.and]

 [aw2.and not_empty="{{aw2.get sc.sms}}"]
   [aw2.sms main="{{aw2.get site_settings.sms-service-provider}}"
            phone_no="{{aw2.get sc.mobile}}" set='sc.sms_result']
   [aw2.get sc.sms /]
   [/aw2.sms]
 [/aw2.and]
[/aw2.shortcode]
```

Code Snippet 1: notification.send shortcode definition

As we can see from above shortcode logic, first we check if there is any “from” e-mail address parameter set, if not use sites admin_email address.

If “to” is not empty which is a to e-mail address parameter passed from the shortcode call, set “to”, “cc”, “from”, “subject” parameters of aw2.wp_mail shortcode. We also, check if there is an attachment to the e-mail which is to be sent. After that set the body for the e-mail and then send the mail using [aw2.wp_mail] shortcode.

Furthermore, to send SMS we check if site setting for sending SMS is set to “on” and if mobile no. exists (to which SMS is to be sent) and also we check if the SMS service provider site setting is set. If these parameters are valid, then send the SMS to the intended recipient.

Shortcode Usage

[notification.send to=’to_email_address’ from=’from_email_address’ // some more parameters]

How shortcodes are handled in the Awesome Studio:

In the Awesome studio, shortcodes are handled slightly in a different way than WordPress. In WordPress add_shortcode function is used to register a shortcode wherein we pass in shortcode name as the first parameter and it’s callable function handler as the second parameter which gets executed when that shortcode is encountered.

Following is syntax of add_shortcode and its attached callable function,

```
function 'shortcode_callable_function' ( $atts ) {
// Logic for short code goes here.

    return "Something from a shortcode function.";
}
add_shortcode( 'foo_shortcode', 'foo_shortcode_callable_function' );
```

Code Snippet 2: add_shortocode syntax.

As we can see from above code snippet ‘foo_shortcode’ is the name of the shortcode and ‘foo_shortcode_callable_function’ is its callable function. This shortcode is generally written in “functions.php” file of the currently activated theme.

In the awesome studio, the shortcode definition is stored in a CPT (Custom Post Type) called “aw2_shortcode”. This CPT’s content (post_content field of WP_Posts table) is actual shortcode definition which is the required logic for that particular shortcode. As per the syntax of aw2.shortcode, the string that comes after is the shortcode name (see syntax of [aw2.shortcode])

Following screenshot depicts this,

Screenshot 1: Shortcodes in Awesome Studio.