

 marketful <small>fulfillment</small>	Documentación del archivo: admin-product-entries.php
	Autor: Mauricio Alcala
	Fecha: 8 de agosto de 2018
	Versión: 1.02
	Autoriza: Adolfo Yanes

Descripción general:

Sección de publicaciones dentro del plugin Marketful que permite modificar metadatos de cada producto dentro de la base de datos de Wp y Wc.

Funciones PHP

```

/*****
* - @Funciones PHP
* Primero se valida con el @parámetro $_POST['nonce'] y si el hash es correcto
* posteriormente manda a llamar a la @función my_theme_ajax_submit()
*
*/
if (isset($_POST['my_theme_ajax_submit']))
if ( wp_verify_nonce( $_POST['nonce'], 'my_theme_ajax_submit' ) )
my_theme_ajax_submit();

/*
***** /

/*****
* - @Función PHP: my_theme_ajax_submit()
* La @función my_theme_ajax_submit() recibe dentro de las variables:
* $producto_id = Identificador del producto
* $value = Nuevo valor a actualizar
* $key = Tipo de metadata que se modificara
*
* Posteriormente despues de recibir los @parámetros ejecuta la función de wp
* update_post_meta(@string,@string,@string) que actualiza la meta data que se envió.
*
* Para finalizar la @función wp_die() finaliza la ejecución y muestra el error
* en caso de que suceda alguno.
*/
function my_theme_ajax_submit() {
// do something
// some php code that fires from AJAX click of #fire
$producto_id = $_POST['product_id'];
$value = $_POST['value'];
$key = $_POST['key'];
### aquí fue
// update_user_meta( 1, "first_name", nombre );
update_post_meta( $producto_id, $key, $value );
// wp_send_json_success([200, "hola"], 200) ; NO SIRVE
wp_die();
}
?>

***** /

```

Funciones JQuery / Ajax

```
/* *****  
* - @Función JQuery/Ajax: cambioStatus(@string,@string)  
* Esta función recibe dos @parámetros que son el id del producto y el tipo de  
* metadato que modificara.  
* La @función console log muestra en la consola del navegador la información del objeto  
* product_id.  
*  
* Se crea una variable con el valor del select que se modificó, se obtiene dicha  
* información obteniendo el id del select con el tipo de metadato que se modificara  
* más guion bajo más el id del producto.  
*  
* Se envía con console.log el valor de key  
* Se crea una función AJAX que pide el tipo de solicitud que se hace 'POST'  
* y se envían los @parámetros además que se manda a llamar a la función de PHP  
* my_theme_ajax_submit. Se le envían los @parámetros: product_id, value y key.  
*  
* La @función Ajax que nombramos response nos responde a la solicitud con un archivo Json.  
*  
* En caso de que la @función haya resultado exitosa, la reflejamos en consola con  
* console.log y lo mostramos en el @boton #fire cambiando su texto a "Cambio  
* correcto".  
* En caso de que la @función Ajax no haya resultado exitosa, reflejamos en  
* consola el error y actualizamos el @boton #fire cambiando su texto a "error".  
*/  
function cambioStatus(product_id, key){  
    console.log(product_id)  
    var value = $('#'+ key + "_" + product_id).val()  
    console.log(key)  
    jQuery.ajax({  
        type: 'post',  
        data: {  
            "my_theme_ajax_submit": "now",  
            "nonce" : "<?php echo wp_create_nonce( 'my_theme_ajax_submit' ); ?>",  
            product_id: product_id,  
            value: value,  
            key: key  
        },  
        success: function(response) {  
            console.log(response)  
            // jQuery('#fire').text("Cambio Correcto!");  
        },  
        error: function(response) {  
            console.log(response)  
            // jQuery('#fire').text("...error!");  
        },  
    });  
};  
***** /
```