

 marketful <small>fulfillment</small>	Documentación del archivo: class-admin.php
	Autor: Mauricio Alcala
	Fecha: 12 de agosto de 2018
	Versión: 1.01
	Autoriza: Adolfo Yanes

Descripción general:

Archivo PHP que tiene la función de hacer la primera llamada a los diferentes elementos que se encuentran dentro de sí.

Funciones PHP

```

/*****
 * @clase PHP MKF_admin
 * @Atributos:
 * private $plg_name (Nombre del plugin)
 * private $plg_small_name (Nombre corto)
 * private $version (version)
 * private $dashboard_menu_name ( Nombre a mostrar en el dashboard)
 *
 */
class MKF_Admin
{
/*****
*** ATRIBUTOS *****/
*****/

    private $plg_name;
    private $plg_small_name;
    private $version;
    private $dashboard_menu_name;

/*****
*** MÉTODOS *****/
*****/

/*****
 * La @función @publica enqueue_styles() sobre escribe el estilo tomado de wp en caso de que no se
 * haya modificado.
 */
    public function enqueue_styles()
    {

        wp_enqueue_style($this->plg_small_name, plugin_dir_url( __FILE__ ).'css/plg-admin.css',
array(),$this->version,'all');
        wp_enqueue_style('datatables',"https://cdn.datatables.net/v/dt/dt-
1.10.18/datatables.min.css",array(), '1.10.18', 'all');
        wp_enqueue_style('category',plugin_dir_url( __FILE__ ) .
'css/category_selection.css',array(), '1.0.0', 'all');
        wp_enqueue_style('iconos',plugin_dir_url( __FILE__ ) . 'css/font-
awesome.min.css',array(), '1.0.0', 'all');
        wp_enqueue_style('bootstrap-map',"https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.0/css/bootstrap.css.map",array(), '4.1.0', 'all');

```

```

    }
    /*****

/*****
* La @función @publica enqueue_styles() sobre escribe los
* @Scripts de JavaScript en caso de que no hayan sido modificados, logra que los datos estén
* disponibles para su secuencia de comandos que normalmente solo puede obtener del lado del
* servidor de WordPress.
*/

public function enqueue_scripts()
{
    wp_enqueue_script('Popper',"https://unpkg.com/popper.js@1.14.3/dist/umd/popper.min.js", array(),
'1.14.3', false
);wp_enqueue_script('bootstrap',"https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min
.js",array(),4.1.0', false );
    wp_enqueue_script('datatables',"https://cdn.datatables.net/v/dt/dt-1.10.18/datatables.min.js" ,
array(), '1.10.18', false );
    wp_enqueue_script('admin_js_bootstrap_hack', plugin_dir_url( __FILE__ ) . 'js/bootstrap-
hack.js', false, '1.0.0', false);
    wp_enqueue_script($this->plg_small_name, plugin_dir_url( __FILE__ ) . 'js/plg-admin.js',
array('jquery'), $this->version, false);
    wp_enqueue_script( 'ajax-script',
        plugins_url( '/js/myjquery.js', __FILE__ ),
        array('jquery')
    );
}
/*****

/*****
* El @Script que a continuación se utiliza la @función wp_create_nonce(@string) que genera un hash
* con esa información para despues ser utilizada para validar la información que se envia.
*/
$title_nonce = wp_create_nonce( 'title_example' );
wp_localize_script( 'ajax-script', 'my_ajax_obj', array(
    // 'ajax_url' => admin_url( 'admin-ajax.php' ),
    'ajax_url' => admin_url( 'admin-ajax.php' ),
    'nonce' => $title_nonce,
) );
}
/*****

/*****
* @Función PHP
* La @función plg_menu() muestra la sección del plugin que corresponda, agregando una página al
* menú y posteriormente manda a llamar al método que muestra dichas secciones.
* Nos muestra *Dashboard, *Publicaciones, *Ordenes, *Preguntas, *Mensajería postventa y
* publicación.
*/
public function plg_menu()
{
    add_menu_page(
        __( $this->plg_small_name, 'textdomain' ),
        __( $this->plg_name, 'textdomain' ),
        $this->available_menu_to_editor_or_admin(),
        $this->dashboard_menu_name,
        array( $this, 'dashboard' ),
        'dashicons-hammer'
    );
}

```

```

add_submenu_page(
    $this->dashboard_menu_name,
    __( "{$this->plg_small_name}-product-entries", 'textdomain' ),
    __( 'Publicaciones', 'textdomain' ),
    $this->available_menu_to_editor_or_admin(),
    "{$this->plg_small_name}-product-entries",
    array( $this, 'entries_view' )
);

add_submenu_page(
    $this->dashboard_menu_name,
    __( "{$this->plg_small_name}-product-orders", 'textdomain' ),
    __( 'Ordenes', 'textdomain' ),
    $this->available_menu_to_editor_or_admin(),
    "{$this->plg_small_name}-product-orders",
    array( $this, 'dashboard' )
);

add_submenu_page(
    $this->dashboard_menu_name,
    __( "{$this->plg_small_name}-product-questions", 'textdomain' ),
    __( 'Preguntas', 'textdomain' ),
    $this->available_menu_to_editor_or_admin(),
    "{$this->plg_small_name}-product-questions",
    array( $this, 'dashboard' )
);

add_submenu_page(
    $this->dashboard_menu_name,
    __( "{$this->plg_small_name}-product-messages", 'textdomain' ),
    __( 'Mensajería Postventa', 'textdomain' ),
    $this->available_menu_to_editor_or_admin(),
    "{$this->plg_small_name}-product-messages",
    array( $this, 'dashboard' )
);

add_submenu_page(
    NULL,
    __( "{$this->plg_small_name}-product-edit", 'textdomain' ),
    __( 'Publicación', 'textdomain' ),
    $this->available_menu_to_editor_or_admin(),
    "{$this->plg_small_name}-product-edit",
    array( $this, 'entries_edit' )
);

}
/*****

/*****

* @Función PHP
* available_menu_to_editor_or_admin()
* Primero se verifica si es una sesión registrada con is_user_logged_in()
* Se define a la variable $is_editor y se le asigna el valor falso
* Después en la variable $user se guarda el valor del objeto usuario que devuelve
* wp_get_current_user()
* Se crea un foreach para recorrer todos los valores (roles) dentro del objeto $user
* Definimos un switch para ver si es administrador o editor. En caso de ser editor se retorna la
* leyenda 'editor' en caso contrario se retorna 'manage:options'
*
*/
public function available_menu_to_editor_or_admin()

```

```

{
    if (is_user_logged_in())
    {
        $is_editor = false;
        $user       = wp_get_current_user();

        foreach ($user->roles as $key => $role)
        {

            switch ($role) {
                case 'administrator': return "manage_options";
                case 'editor': $is_editor = true;
                break;
                default: break;
            }
        }

        if ($is_editor) { return "editor"; }

    }
    return "manage_options";
}
/*****

/*****
* @Función dashboard()
* Método que manda a llamar el archivo admin-dashboard.php
*/
public function dashboard() {
    include_once "partials/admin-dashboard.php";
}
/*****

/*****
* @Función entries_view()
* Método que manda a llamar el archivo admin-product-entries
*/
public function entries_view() {
    include_once "partials/admin-product-entries.php";
}
/*****

/*****
* @Función entries_edit()
* Método que manda a llamar el archivo admin-product-edit-form.php
*/
public function entries_edit() {
    include_once "partials/admin-product-edit-form.php";
}
/*****

```