marketful fulfillment	Documentación del archivo: class-core.php
	Autor: Mauricio Alcala
	Fecha: 13 de agosto de 2018
	Versión: 1.01
	Autoriza: Adolfo Yanes

Descripción general:

Crea la clase MKF_Core y define valores para diferentes atributos, además define los hooks públicos del plugin.

Funciones PHP

```
/**********************************************************************************************************
 * @Script
 * Se valida que exista la clase MKF_Core en caso de que sea
 * falso continua.
 */
if ( !class_exists( 'MKF_Core', false ) ):
/**********************************************************************************************************



/**********************************************************************************************************
 * @Clase MKF_Core
 *
 * @atributos
 * -Creamos variables @protected $loader, $plugin_id, $version,
 * $plugin_root.
 *
 */
class MKF_Core
{
  protected $loader;
  protected $plugin_id;
  protected $version;
  protected $plugin_root;
/**********************************************************************************************************/



  /**********************************************************************************************************
   * @función __construct()
   *
   * Dentro de esta función primeramente asignamos los valores
   * de los diferentes atributos de la clase.
   */
  public function __construct()
  {
    $this->version    = defined( 'PLUGIN_GVERSION' ) ? PLUGIN_GVERSION : '1.0.0';
    $this->plugin_id  = defined( 'PLUGIN_GSNAME' )   ? PLUGIN_GSNAME   : 'mkf';
    $this->set_root_path();
    $this->load_dependencies();
    $this->set_locale();
    $this->define_admin_hooks();
    $this->define_public_hooks();
  }
  /**********************************************************************************************************/



  /**********************************************************************************************************
   * @función set_root_path()
   *
   * Se asigna el valor de dirname(__FILE__) al atributo
   * plugin_root que le devuelve el valor del nombre en el
```

```
 * directorio del Path.
 */
public function set_root_path()
{
  $this->plugin_root = plugin_dir_path( dirname( __FILE__ ) );
}
/*****************************************************************************************/



/*****************************************************************************************
 * @función load_dependencies()
 *
 * Se manda a llamar a las siguientes clases y se crea un nuevo objeto
 * llamado loader del tipo MKF_Loader().
 */
private function load_dependencies()
{
  require_once $this->plugin_root . 'includes/class-loader.php';
  require_once $this->plugin_root . 'includes/class-i18n.php';
  require_once $this->plugin_root . 'includes/class-dbcore.php';
  require_once $this->plugin_root . 'admin/class-admin.php';
  require_once $this->plugin_root . 'admin/class-admin-product-entries.php';
  require_once $this->plugin_root . 'admin/class-admin-wc-proxy.php';
  $this->loader = new MKF_Loader();

}
/*****************************************************************************************/



/*****************************************************************************************
 * @función set_locale()
 *
 * Se crea una @función provada set_locale() que crea un objeto llamado
 * $i18n del tipo MKF_i18n().
 *
 * Se agrega una acción al objeto Loader mandando a llamar
 * plugins_loaded, usando el objeto $i18n y la funcion
 * load_pluginn_textdomain
 */
private function set_locale()
{

  $i18n = new MKF_i18n();

  $this->loader->add_action( 'plugins_loaded', $i18n, 'load_plugin_textdomain' );

}
/*****************************************************************************************/



/*****************************************************************************************
 * @funcion define_admin_hooks()
 *
 * Dentro de esta función definimos un nuevo objeto llamado $plugin_admin
 * del tipo MKF_Admin, así como a otro producto llamado $product_entry
 * Despues se agregan 3 acciones que son poner en cola los estilos,
 * despues se ponen en cola los scripts y por ultimo se agrega la accion
 * de admin_menu y el metodo plg_menu.
 *
 * CRUD ProductEntries (Definir la accion para procesar un formulario html
 * con method POST)
 */
private function define_admin_hooks()
{
  $plugin_admin = new MKF_Admin( $this->get_plugin_small_name(), $this->get_version() );
  $product_entry = new MKF_ProductEntry( $this->get_plugin_small_name(), $this->get_version() );

  $this->loader->add_action( 'admin_enqueue_scripts', $plugin_admin, 'enqueue_styles' );
  $this->loader->add_action( 'admin_enqueue_scripts', $plugin_admin, 'enqueue_scripts' );
  $this->loader->add_action( 'admin_menu',            $plugin_admin, 'plg_menu' );
```

```
  $this->loader->add_action(
    'admin_post_add_metadata_to_product_entry',
    $product_entry,
    'prefix_admin_add_metadata_to_product_entry'
  );
}
/*************************************************************************************************/



/*************************************************************************************************/
 * @función define_public_hooks()
 */
private function define_public_hooks()
{

}
/*************************************************************************************************/



/*************************************************************************************************
 * @función run()
 * Se asigna al atributo loader el valor que retorna el @método run()
 */
public function run()
{
  $this->loader->run();
}
/*************************************************************************************************/



/*************************************************************************************************
 * @función get_plugin_small_name()
 * Se regresa el valor del atributo plugin_id
 */
public function get_plugin_small_name()
{
  return $this->plugin_id;
}
/*************************************************************************************************/



/*************************************************************************************************
 * @función get_loader()
 * Se regresa el valor del atributo loader
 */
public function get_loader()
{
  return $this->loader;
}
/*************************************************************************************************/



/*************************************************************************************************
 * @función get_version()
 * Se regresa el valor del atributo version
 */
public function get_version()
{
  return $this->version;
}
/*************************************************************************************************/
```