

Manual del plugin de Prestashop para [Sales Layer](#)

Versión 1.4

El plugin de Sales Layer para Prestashop se encarga de actualizar la información de producto de una o diversas tiendas de Prestashop desde uno o varios conectores configurados en Sales Layer.

El plugin está disponible para su descarga en el **cloud Sales layer** en la configuración de canal Prestashop y en [GitHub](#) para realizar modificaciones personalizadas.

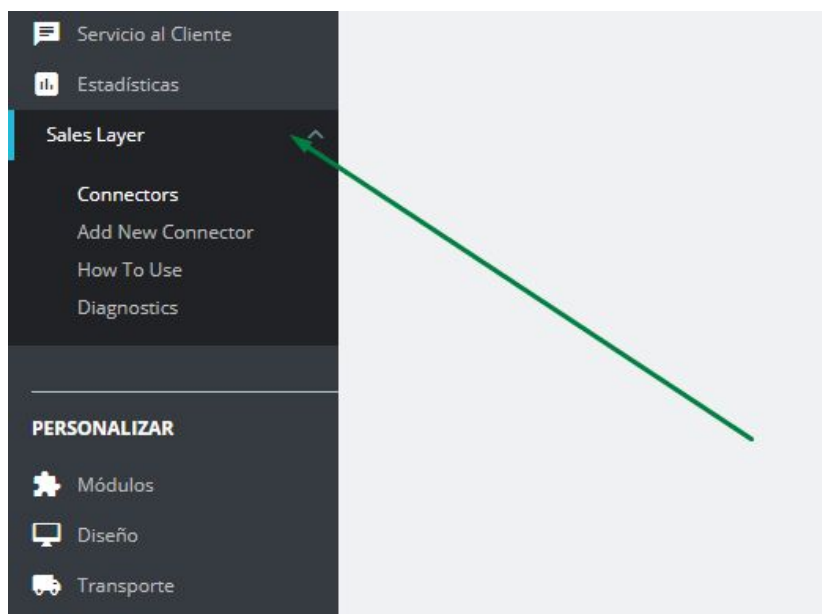
Instalación:


Atención:

Si se ha usado una versión anterior de nuestro plugin, es recomendable desinstalarla con el checkbox para eliminar la carpeta del plugin. Luego se podrá instalar la nueva versión asegurándonos que todo estará correcto.

Desde la página de Admin en la tienda de Prestashop accede menú izquierdo y elige la opción **Módulos > Module Manager > Subir un Módulo** y carga el zip del plugin descargado.

Si todo ha ido bien, presionando F5 (refrescar página) debería aparecer [Sales Layer](#) en el menú izquierdo.



En la opción **How To Use** Podemos encontrar un pequeño manual que nos ayudará a entender, y también a comprobar, si tenemos todo lo necesario para correcto funcionamiento del plugin. Si tenemos todos los requisitos necesarios aparecerán el icono ().

Todos los elementos marcados son necesarios para el correcto funcionamiento del plugin.

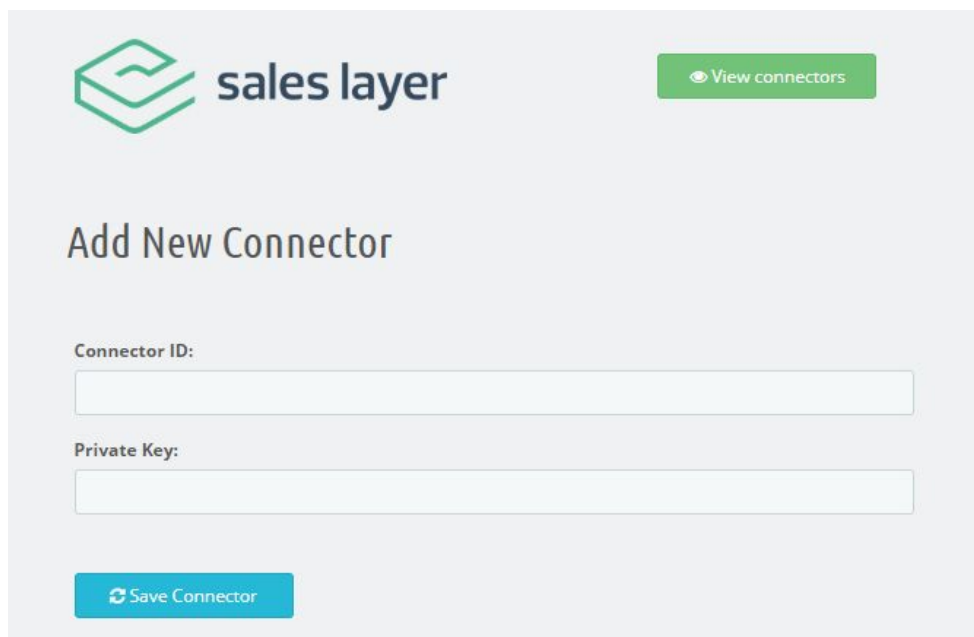
En caso de que “Registered prestashop cronjob activity” aparezca marcado como un error, se nos estará indicando que en la última hora no ha sido detectada la actividad de CRON de Prestashop. En tal caso es posible que haya que acceder al servidor (*vía SSH o cpanel*) tal como se explica en la parte:

HOW TO SYNCHRONIZE BY CRON.

Si el plugin ya está activo, puede esperarse un tiempo hasta que se ejecute el CRON por lo menos una o dos veces para que se pueda calcular la frecuencia de ejecuciones del CRON y el mensaje de error desaparecerá.

Si han ocurrido algunos errores es posible que haya avisos en los archivos de diagnóstico en el **Diagnostics**. Si se tratase del mismo error, pueden eliminarse los archivos para que empiece con contenido limpio antes de sincronizar.

Primeros Pasos:



Para agregar un nuevo conector debemos acceder a la pestaña **Connectors** y luego hacer click en **+ New connector**.

Para conectar el plugin con Sales Layer solo hay que copiar y pegar las credenciales del conector configurado en Sales Layer para Prestashop y continuar presionando el botón **Save connector**.

Si el ID y clave privada son correctas aparecerá el nuevo conector en la página de conectores.

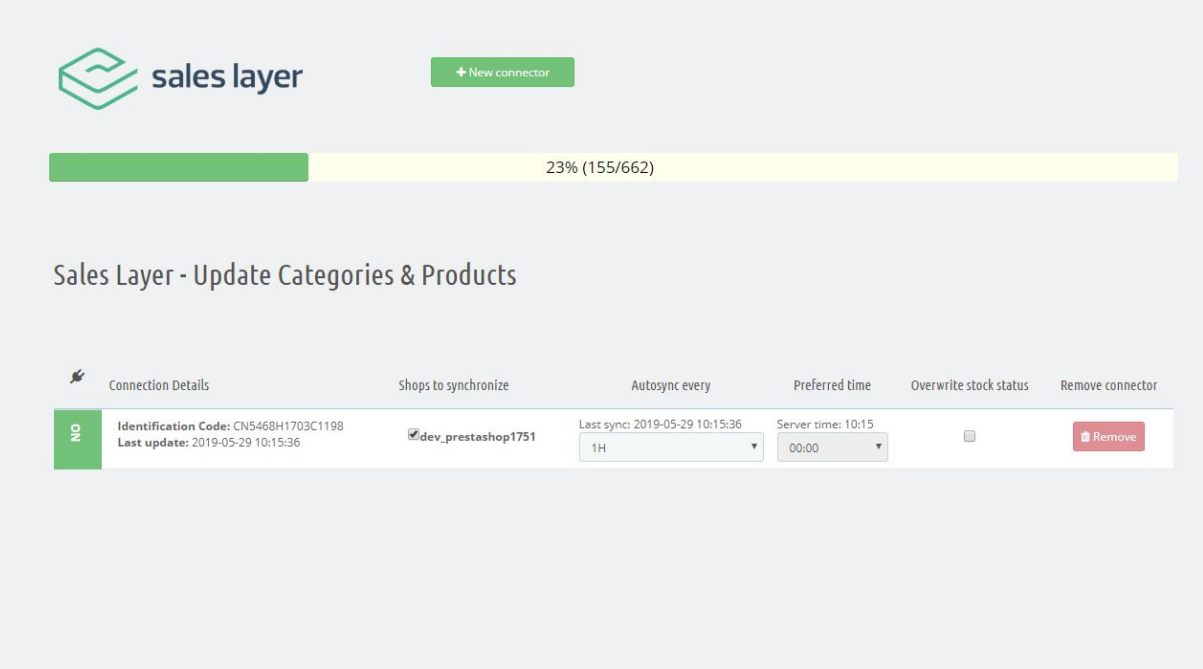
El plugin necesita, para su funcionamiento automático, que el **Prestashop cronjobs** esté **activo** y funcionando. La frecuencia de ejecución de los cronjobs recomendada es de 5 minutos. No obstante, si ya se están usando para otras tareas con un periodo de entre 1-30 minutos, puede mantenerse en esos tiempos. El plugin automáticamente ajusta el tiempo de un proceso para la sincronización, calculará la diferencia entre cada ejecución de cron, y si no tiene demasiada carga de CPU, se llamara de nuevo para seguir sincronizando sin detenerse.

En el caso de que el servidor tenga demasiada carga no se autoejecutará, pero continuará con la sincronización cuando de nuevo se ejecute el cron (*en segundo plano para que el servidor no sufra demasiada carga por procesos de sincronización; la web seguirá funcionando de forma fluida*).

Para tener un conector activo basta con elegir en "Select" la frecuencia con la que queremos que se ejecute. Si el valor supera 24 h se desbloquea el "Select" de horario preferido para la sincronización, donde podemos configurar si queremos que se

ejecute, por ejemplo, a las 3 de la mañana, cuando termine la jornada de los editores, etcétera (se sincronizará con todos los cambios realizados en el cloud desde la última sincronización).

Sincronización:



The screenshot shows the Sales Layer interface. At the top, there is a progress bar indicating 23% completion (155/662). Below this, the title "Sales Layer - Update Categories & Products" is displayed. A table lists the connectors, with columns for Connection Details, Shops to synchronize, Autosync every, Preferred time, Overwrite stock status, and Remove connector.

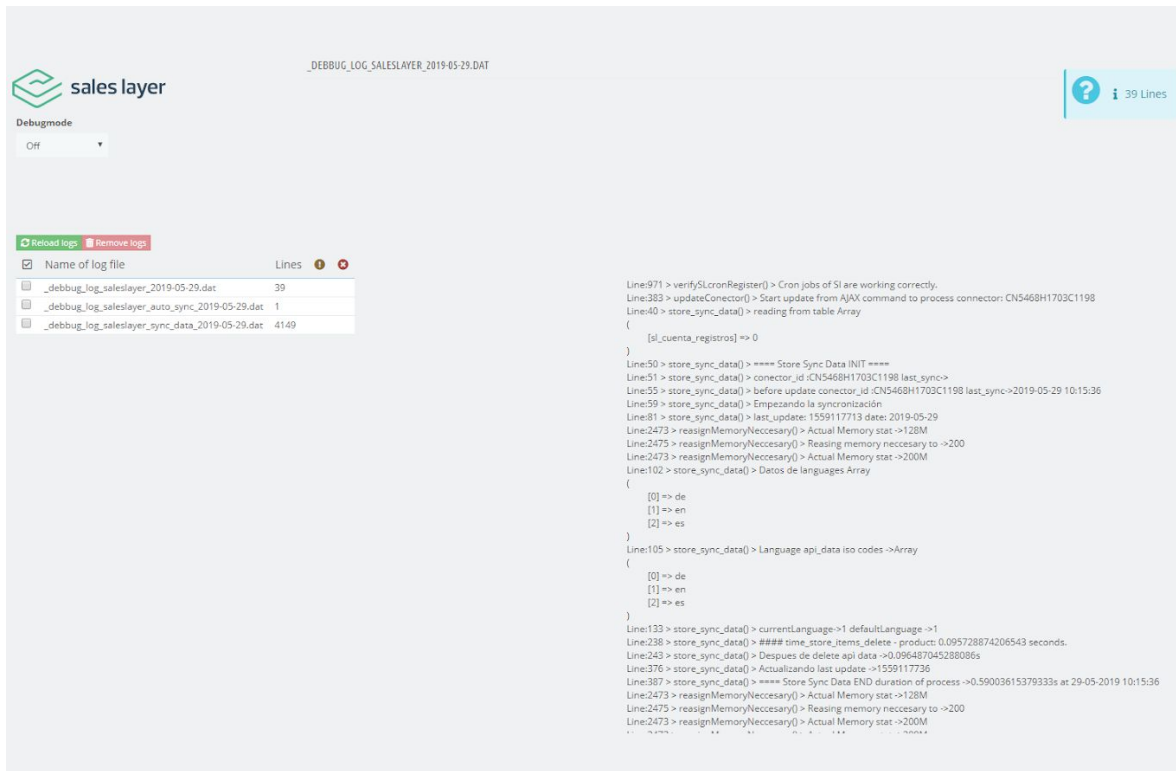
	Connection Details	Shops to synchronize	Autosync every	Preferred time	Overwrite stock status	Remove connector
NO	Identification Code: CN5468H1703C1198 Last update: 2019-05-29 10:15:36	<input checked="" type="checkbox"/> dev_prestashop1751	Last sync: 2019-05-29 10:15:36 1H	Server time: 10:15 00:00	<input type="checkbox"/>	Remove

Cada vez que el conector reciba nuevas actualizaciones de productos por sincronizar, los almacenará en la base de datos temporalmente y seguirá procesando todo su contenido de forma asíncrona, buscando ocupar la menor cantidad de memoria y procesamiento de CPU.

Para reconocer si todavía está sincronizando podemos acceder al apartado de **Connectors** y, si todavía tiene registros por procesar, aparecerá la barra de progreso con los elementos totales pendientes y el porcentaje de progreso.

Si el proceso está detenido puede aparecer un mensaje indicando en cuanto tiempo se ejecutará el CRON para reanudar la sincronización y continuará con ella. Si ha ocurrido algún error con algún elemento, intentará sincronizar 2 veces más y creará un mensaje en un archivo de diagnóstico indicando qué problema ha tenido lugar y, siempre que sea posible, cómo solucionarlo.

Diagnóstico y solución de errores



The screenshot shows the Sales Layer debug interface. At the top, there's a header with the Sales Layer logo and a title bar. Below the header, there's a "Debugmode" section with a dropdown menu set to "Off". To the right, there's a help icon and a notification "39 Lines". Below the debug mode section, there's a "Reload logs" button and a "Remove logs" button. A table lists log files with columns "Name of log file" and "Lines". The table shows three log files: "_debug_log_saleslayer_2019-05-29.dat" (39 lines), "_debug_log_saleslayer_auto_sync_2019-05-29.dat" (1 line), and "_debug_log_saleslayer_sync_data_2019-05-29.dat" (4149 lines). To the right of the table, there's a detailed log entry for the third file, showing various debug messages and data structures.

```

Line:871 > verifySLcronRegister() > Cron jobs of SL are working correctly.
Line:883 > updateConnector() > Start update from AJAX command to process connector: CN5468H1703C1198
Line:40 > store_sync_data() > reading from table Array
(
  [sl_cuenta_registros] => 0
)
Line:50 > store_sync_data() > ===== Store Sync Data INIT =====
Line:51 > store_sync_data() > connector_id <CN5468H1703C1198 last_sync=>
Line:55 > store_sync_data() > before update connector_id <CN5468H1703C1198 last_sync=>2019-05-29 10:15:36
Line:59 > store_sync_data() > Empezando la sincronización
Line:81 > store_sync_data() > last_update: 1559117713 date: 2019-05-29
Line:2473 > reassignMemoryNecesary() > Actual Memory stat ->128M
Line:2475 > reassignMemoryNecesary() > Reasing memory necessary to ->200
Line:2473 > reassignMemoryNecesary() > Actual Memory stat ->200M
Line:102 > store_sync_data() > Datos de languages Array
(
  [0] => de
  [1] => en
  [2] => es
)
Line:105 > store_sync_data() > Language api_data iso codes ->Array
(
  [0] => de
  [1] => en
  [2] => es
)
Line:133 > store_sync_data() > currentLanguage->1 defaultLanguage ->1
Line:238 > store_sync_data() > ### time_store_items_delete - product: 0.095728674206543 seconds.
Line:243 > store_sync_data() > Despues de delete api data ->0.096487045288086s
Line:376 > store_sync_data() > Actualizando last update ->1559117736
Line:387 > store_sync_data() > ===== Store Sync Data END duration of process ->0.59003615379333s at 29-05-2019 10:15:36
Line:2473 > reassignMemoryNecesary() > Actual Memory stat ->128M
Line:2475 > reassignMemoryNecesary() > Reasing memory necessary to ->200
Line:2473 > reassignMemoryNecesary() > Actual Memory stat ->200M

```

En la opción **Diagnostics** tenemos disponible una herramienta para listar los archivos de diagnóstico generados por nuestro conector. Inicialmente hay varias opciones a la hora de ejecutar en el modo “debug” en **Off**. En este modo solo se crean archivos si ocurren errores e imprimirá solo los textos de debug de un elemento en el cual haya ocurrido un error.

En otras opciones de debug (*subiendo de “level”*) aparecerá más información acerca del uso de memoria y CPU. Esta información se pueden usar para el desarrollo y solución de problemas. Por defecto, no obstante, la mejor opción es tenerlo en modo **Off** y así ver solo si ocurren errores para no crear archivos de gran tamaño.

Los archivos de diagnóstico se deberían eliminar de forma automática tras transcurrir 15 días. Los errores que aparecen en el archivo estarán marcados con un icono y en fondo rojo. Los ítems con error aparecerán con un mensaje informativo y la referencia identificativa de Sales Layer para poder ir a revisar la información si se requiere. Un ejemplo de mensaje sería:

Error. Creating category ID: 91

Con este mensaje sabemos que se trata de una categoría y que su referencia en Sales Layer es “91”.

En caso de que el error no tenga una referencia identificativa es posible que sea porque haya ocurrido en algún punto intermedio del código y, en este caso, lo podremos identificar en el archivo de log general que empieza por:

`_debug_log_saleslayer_sync_data_`

...seguido con la fecha del día de lo ocurrido. En el archivo podremos encontrar más información del elemento en el que ha ocurrido el problema (*presionando en el icono de error*) y podremos saltar de un error a otro y navegar entre los errores (*si se han producido más de uno*).

Los errores que pudieran encontrarse antes de la sincronización de un ítem, muy probablemente tengan relación con este hecho. Con ello podremos identificar si se trata de una categoría o otro elemento, y qué referencia concreta tiene para revisarlo en Sales Layer.

Rendimiento

La velocidad de sincronización depende de muchos factores como, por ejemplo, la número de ítems y cantidad de datos que se necesitan para completar la sincronización, cuantas imágenes se han de subir y cómo de grandes son, la carga de proceso existente en el servidor, la potencia de las bases de datos, etcétera... El plugin está configurado para que puede calcular la memoria que puede necesitar durante este proceso. Al sincronizar, el mínimo que se le asigna por defecto son 200MBytes, pero en el caso de que, después de descargar muchos datos, le quede poco espacio de memoria, se auto-asignará más para poder terminar todo el proceso. En modo de test, el uso promedio de memoria es de 15MBytes, pero con un límite máximo de seguridad establecido en 200MBytes.

En diferentes pruebas, la sincronización ha arrojado estos resultados relativos:

categorías : 0.183 - 0.583 segundos
productos : 0.185 - 1.085 segundos
variantes : 0.357 - 0.550 segundos

```
pid:30899-mem:12.00-cpu:0.92-time:96-from:[saleslayerimport.php] Line:2068 > update_items() > #### time_sync_stored_category: 0.29174995422363 seconds.  
pid:30899-mem:12.00-cpu:0.92-time:96-from:[saleslayerimport.php] Line:2068 > update_items() > #### time_sync_stored_category: 0.1877760887146 seconds.  
pid:30899-mem:12.00-cpu:0.92-time:96-from:[saleslayerimport.php] Line:2068 > update_items() > #### time_sync_stored_category: 0.18359899520874 seconds.  
pid:30899-mem:14.00-cpu:0.92-time:96-from:[saleslayerimport.php] Line:2106 > update_items() > #### time_sync_stored_product: 0.18540215492249 seconds.  
pid:30899-mem:14.00-cpu:0.92-time:96-from:[saleslayerimport.php] Line:2154 > update_items() > #### time_sync_stored_product_accessories: 0.0042471885681152 seconds.  
pid:30899-mem:14.00-cpu:0.93-time:97-from:[saleslayerimport.php] Line:2133 > update_items() > #### time_sync_stored_product_format: 0.5506329536438 seconds.  
pid:30899-mem:14.00-cpu:0.93-time:98-from:[saleslayerimport.php] Line:2133 > update_items() > #### time_sync_stored_product_format: 0.55842781066895 seconds.  
pid:30899-mem:14.00-cpu:0.93-time:98-from:[saleslayerimport.php] Line:2133 > update_items() > #### time_sync_stored_product_format: 0.35768008232117 seconds.
```

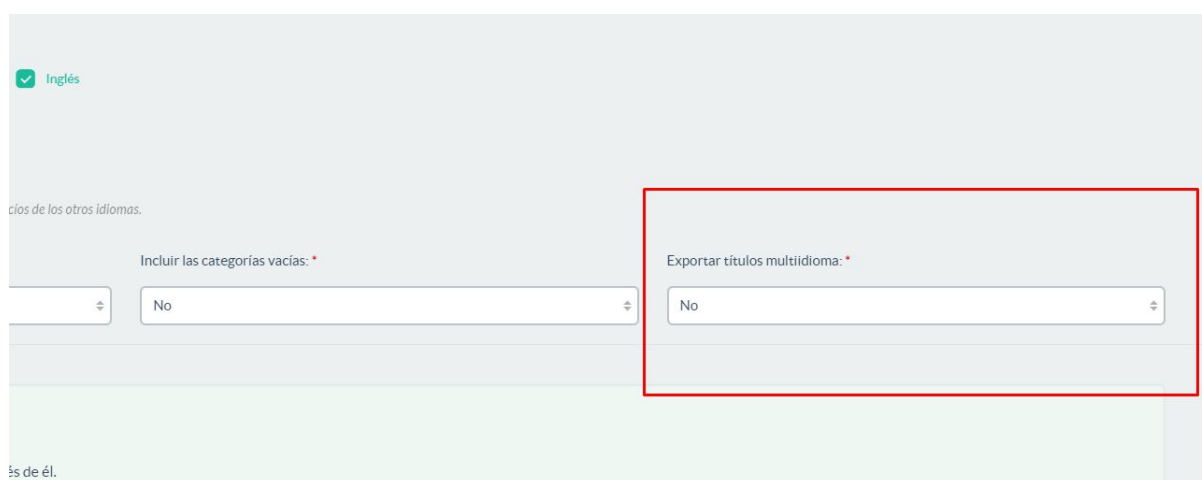
Para poder ver los tiempos se ha de acceder al plugin en *modo debug* y localizar el archivo generado que empieza por `_debug_log_saleslayer_timers_{fecha del día}`. Aquí podremos ver los tiempos que ha durado cada ítem en procesarse.

Configuración extra de conector en el cloud:

Exportar títulos multi-idioma:

Entre las novedades que han sido agregadas en la actual versión se encuentra la funcionalidad de exportar títulos multi-idioma, y para eso tenemos que tener activado **Campos multi-idioma** en Sales Layer y configurarlos en el conector de sincronización.

En el conector de Prestashop en Sales Layer nos deberá aparecer la opción **“Exportar títulos multi-idioma”**:

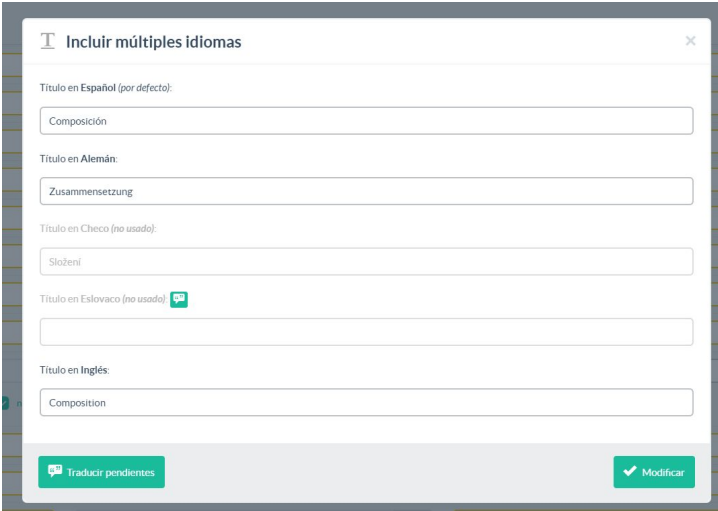


(Si no aparece esa funcionalidad en tu cuenta de Sales Layer puedes contactar con el servicio de soporte de Sales Layer para pedirles que la activen)

Después de activar *“Exportar títulos multi-idioma”* con la opción **“SÍ”**, nos hemos de mover a la pestaña **“Datos De salida”** y comprobar que cada campo tiene el título en varias idiomas. Desde esa misma opción podemos editar los diferentes títulos si es necesario:



Al hacer click sobre el botón de los títulos traducidos, nos permitirá editar los títulos que deseamos para cada idioma:



Incluir múltiples idiomas

Título en Español (por defecto):
Composición

Título en Alemán:
Zusammensetzung

Título en Checo (no usado):
Složení

Título en Eslovaco (no usado):
[Icono de traducción]

Título en Inglés:
Composition

Traducir pendientes Modificar

Durante la sincronización se asignan esos nombres al atributo o característica a la que pertenecen. Después de la sincronización podremos ver que los valores se crean en nuestra tienda de prestashop:



* Nombre: Zusammensetzung de

Creación de características del producto y atributos de variante

El conector de Sales Layer nos permite configurar la exportación de forma que todos los campos adicionales que se agreguen en productos se conviertan en características del producto, y que todos los campos adicionales agregados a la plantilla de variantes se conviertan en atributos configurables de las variantes. **Si el valor está nulo o vacío, se eliminará del producto o variante en su tienda.**

Los nombres primarios de los campos configurados en el conector de Sales Layer se buscarán en Prestashop y, de no existir, se crearán como nuevos:



Campos de atributos:

Normal Color Color

Incluir campo nuevo Incluir múltiples campos

A partir de la versión 1.7 del plugin, este cuenta con posibilidad enviar features con más de un valor. Para separarlos puede usarse el símbolo “[]”.

Para crear las características de tipo personalizadas se puede añadir al final del valor del campo “:custom”, con una fórmula o en el valor del formulario. Para crear todas las características de la misma manera sin tener que enviar un comando custom, es posible editar el archivo saleslayerimport.php y editar la línea que dice:

```
public $create_new_features_as_custom = false;
```

y cambiar el valor a **true**.

- Las características y atributos y sus valores en multiidioma **se crean solo si no existen en la tienda**.
- Si un valor en alguno de las idiomas ya existe, se vincula con el producto y rellena los idiomas faltantes. Si ninguno de los valores ha sido encontrado en la tienda en alguno de los idiomas se crea como nuevo.
- Si la primera vez que se crean los atributos se envían datos erróneos del idioma, no será posible editarlos desde el plugin. Para proteger los valores existentes en caso de que un usuario envíe en un producto con datos erróneos, se editará el idioma existente con un valor erróneo en todos los productos de la tienda, por eso el plugin no edita valores rellenos solo los crea si no existen.
- Es posible editar los idiomas desde Prestashop o eliminarlos, así la siguiente ocasión se crearán ¡ datos correctamente.

Creación de Campos personalizados

Para crear campos personalizados podemos usar el campo **Customizable** en la configuración del conector en el cloud de Sales Layer. El plugin reconoce comandos internos que podemos añadir detrás de frase que ponemos como nombre para el campo que aparece detrás de campo **"TU TEXTO PARA CAMPO:file"** o **"TU TEXTO PARA CAMPO:archivo"**. Comando para que el campo se crea con la configuración que obliga a cliente a rellenar el campo deseado **"TU TEXTO PARA CAMPO:required"** o **"TU TEXTO PARA CAMPO:requerido"**. Los campos podemos crear como texto separado por comas, lista de atributos o relleno con una fórmula como texto separado por coma. Si deseamos enviarlo en varias idiomas debemos elegir el campo de texto.

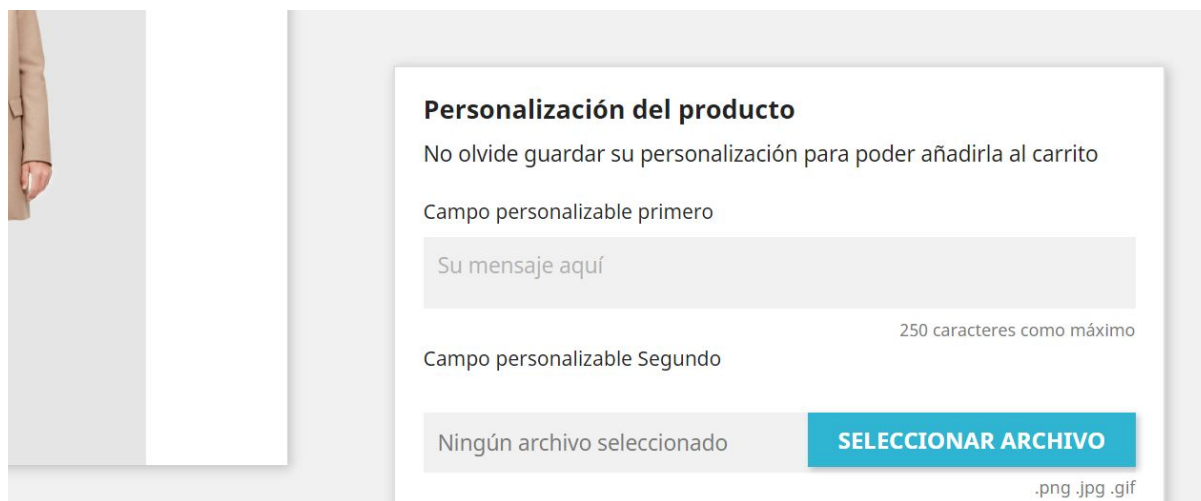
Ejemplo:

Mi Campo Personalizado:file, Mi otro campo personalizado:required, Mi campo para imagenes:file:required

que creará campos en prestashop en el producto:

- Mi Campo Personalizado - Campo para subir imagenes.

- Mi otro campo personalizado - Campo de texto que tendrá configuración obligatoria rellenar por el cliente.
- Mi campo para imágenes - Campo para subir imagenes obligatorio rellenar al enviar el pedido.



Colores personalizados

El plugin contiene desde la versión 1.4.3 una carpeta llamada **colors** que contiene una pequeña base de datos de colores que reconoce por cada idioma. Esta carpeta se usa para reconocimiento de atributos de colores inexistentes en prestashop para los variantes. Significa que si de Sales layer viene un variante con color **verde claro** en español el plugin intentará encontrar el archivo **es.txt** elimina acentos y espacios buscará el color **verdeclaro** y le asigna un color **#90ee90** que tiene predefinido en el archivo. Estos archivos se pueden personalizar para sus necesidades. La razón es que cada empresa usa distintos nombres, distinta mapa de colores en varias idiomas para reconocer los colores. Esto le permite tener siempre su archivo y con cada actualización simplemente copia y pega sus archivos de colores en el plugin. Es suficiente tener un archivo en una idioma que vamos a usar para reconocer colores y asignarlos la primera vez que se creara el atributo.

Otra forma de enviar color es que se envíe el color como texto de manera:

verde claro:#90ee90 El plugin acepta todo antes de **#** como nombre y todo después de : como color **#90ee90**.

Tags

Desde la versión del plugin 1.4.7 ha sido agregada la posibilidad de gestionar los tags de productos. Si no desea que el conector edite o elimine los Tags existentes tiene la posibilidad de deshabilitar el campo **Tags** en la configuración del conector.

Adjuntar archivos

Adjuntar archivos "**File attachment**" está disponible desde la versión 1.4.7 con el nombre del campo **File attachment** en la configuración del conector en el cloud de sales layer .

Importante:

Si no se desea que el conector edite o elimine archivos existentes que se encuentren en Prestashop vinculados con el producto, hay que desactivar este campo en el conector. Si no se desea gestionarlos desde el plugin, en el momento en que el conector detecte que el archivo no recibe desde Sales Layer, se procede a la eliminación del producto en su tienda de prestashop. El campo no soporta multiidioma, solo archivos adjuntos que se asignarán al producto.

Gestión de descuentos

En la versión 1.4.7 ha sido mejorada la gestión de descuentos para productos.

Importante:

Si no se desea que el plugin elimine descuentos actuales de Prestashop, puede desactivar los campos **product_discount_1** y **2** en el cloud de sales layer.

Hay varias posibilidades de crear descuentos, con un máximo de 2 descuentos por producto.

Si se desea crear descuentos para todas las tiendas desde un conector, es posible adjuntar al campo **product_discount_1_type** o **product_discount_2_type** el comando **:all** o **:global**

que el plugin reconoce como crear descuento para todas las tiendas.

Un ejemplo : **percentage:all** o **%:global**

Importante:

Si va a usar descuentos para todas las tiendas, el comando formateará todos los descuentos de todas la tiendas, y dejará solo los descuentos que recibe del conector y los creará para todas las tiendas. Si va a usar descuentos desde un conector con comando **:all** hay que deshabilitar los campos de descuentos en otros conectores, porque cada conector podría crear descuentos que el conector con el comando **:all** eliminaría.

No hay que olvidar que si se necesita tener descuentos diferentes por cada tienda, no se puede usar comando **all** o **global** en ninguno de los conectores. El comando **all** solo debe ser usado para gestionar descuentos en todas las tiendas desde un conector.

Valores permitidos por los campos de descuentos:

product_discount_1 -> numérico 1-100 si el campo **product_discount_1_type** está configurado como descuento por porcentaje. Para eliminar todos descuentos envíe esta línea vacía. Para no editar descuentos desactive estos campos desde conector en el cloud.

Si el campo **product_discount_1_type** está configurado por cantidad, puede poner la cantidad que se debe descontar en el campo **product_discount_1**.

product_discount_1_type -> Permitidos para cantidad (\$, euro, €, dollar, amount, importe)
Permitidos para porcentaje (% , porcentaje, percentage)
Si no está relleno valor por defecto es cantidad.

Ejemplo para adjuntar el comando all -> **porcentaje:all**

product_discount_1_quantity -> Número a partir de cuantas unidades es aplicable el descuento. Si no está relleno el valor por defecto es 1.

Transporte

Para elegir un transporte para el producto, debe crearse desde prestashop y desde Sales Layer enviar su **nombre** o **id_reference** en el campo **Carrier** en la pestaña de productos. Si desea puede deshabilitar el campo para que no elimine transporte existentes en su tienda.