
PrestaShop Beslist.nl Shopping Cart Integration Documentation

Release 1.5

Mark Wienk

Oct 25, 2018

Contents:

1	Installation	3
----------	---------------------	----------

The Beslist Shopping cart integration module is a module for PrestaShop. The module is available for sale at the official Prestashop addons store.



Fig. 1: PrestaShop Beslist integration

Buy the module via the [Prestashop addons store](#). When your PrestaShop instance is connected to your addons account you can now find the module in your backoffice.

You can also download the zip file from the addons website and upload it to your website via the interface. Managing modules works via your Prestashop administration, via the menu *Modules* > 'Modules & Services'. At the top right is a 'Add new module' button, where you can upload the zip file.

After successful installation you can see the [Module configuration page](#).

1.1 Beslist.nl configuration

The Beslist.nl configuration has to be setup at your [Beslist.nl account](#)

1.1.1 Shopping cart settings

The module supports 1 way of calculating shipping costs, namely the option 'The order costs are calculated on the basis of the total turnover of the order (s)'. Set this as an option on Beslist.nl (for both Belgium and the Netherlands). You do this on the page *Shopping cart* -> 'Settings'. Make sure the specification matches the calculation of the shipping costs within your Prestashop installation. You can choose to create a new carrier, where the settings correspond exactly to the settings on Beslist.

1.1.2 API keys

The module uses multiple Beslist API's. Namely:

1. Order API
2. Shopitem API

For this it is necessary to obtain API keys. You will receive these from Beslist via email.

1.1.3 Test API

Initially, a key is provided for the test environment of the Shop Item API. The Order API always uses the production environment, but can be tested in a different way.

After you have made a number of successful Shop Item API notifications, you can receive a production API key from Beslist.

Note: the module can only be put into test mode in its entirety, if you want to use the Order API already, keep in mind that the Shopitem API does nothing.

1.2 Module configuration

The module configuration can be found in your Prestashop installation. Navigate to *Modules* -> 'Modules and Services', search for the *Beslist Shopping cart integration* module and click on 'Configure'.

1.2.1 Configure the test connection

It is highly recommended to first configure a test connection. From Beslist you (if it is correct) have received the data for a test connection. Even when you are running in another system, it is important to first use the Test API keys.

Make sure you use the Test API keys for the test connection

1.2.2 Configuration options

Here you will find a brief explanation for each configuration option.

Beslist productfeed settings

The settings under the heading 'Decisive product feed settings' only apply to the product feed functionality of the module.

- *Standard size attribute*: If your products use different sizes, you can enter it here.
- *Standard color attribute*: If your products use different colors, you can enter it here.
- *Use longe descriptions in feed*: If *YES* your long description is passed to Beslist, by default the feed uses the short product description.
- *Beslist productreference field*: Here you indicate which field is used as a unique field. If it is good, you have passed it on Decidedly. By default, the module uses a combination of the Prestashop ID of the product and the product combination ID (this field is also used in the feed, and works with Channable). The other options are EAN-13 (barcode) and Productreference (your internal reference). In most cases the standard works, so always try first.
- *Filter products without stock*: Do not include products in the feed that do not have inventory.

Beslist.nl settings

These settings only apply to the publications on Beslist.nl, the Dutch website.

- *Use Beslist.nl*: Synchronize orders and products with Beslist.nl (Netherlands).

- *Carrier*: Select a carrier for your Dutch orders. The order and transport costs must match (on Beslist.nl and in your Prestashop carrier configuration).
- *Delivery period*: The delivery time for Dutch orders for which the products are in stock.
- *Delivery period no stock*: The delivery time for Dutch orders for which the products are not in stock.

Beslist.be settings

These settings only apply to the publications on Beslist.be, the Belgian website.

- *Use Beslist.be*: Synchronize orders and products with Beslist.be (Belgium).
- *Carrier*: Select a carrier for your Belgian orders. The order and transport costs must match (on Beslist.be and in your Prestashop carrier configuration).
- *Delivery period*: The delivery time for Belgian orders for which the products are in stock.
- *Delivery period no stock*: The delivery time for Belgian orders for which the products are not in stock

Beslist Shopping cart settings

The settings under the heading ‘Beslist shopping cart settings’ apply to both Beslist.nl and Beslist.be.

- *Enable Beslist Shopping cart*: Enable or disable the Beslist shopping cart functionality for this website
- *Shop ID*: The Shop ID you received from Beslist.
- *Client ID*: The Client ID that you received from Beslist.
- *Order API key*: The personal Order API key that you received from Beslist.
- *Shopitem API key*: The key you received for the Shop Item API.
- *Gebruik testverbinding*: Activate the Test APIs
- **Test productnummer**: Here you fill in the code (‘\$combinationID-\$productid’, EAN-13 or product reference) of the product
 - **Standard**: a combination of the product and combination ID of the product (#combination-#product). Example:
 - * 3-4 Product ID: 4, Combination ID: 3 (visible in the url and database, ‘id_product’ & id_product_attribute)
 - * 4 Product ID: 4, no combination (visible in the url and database, ‘id_product’)
 - EAN-13: The EAN code of the product (visible on the product page)
 - Product reference: The product reference of the product (visible on the product page)
- *From date*: The date from when orders are to be synchronized. This field is automatically updated after a synchronization.
- *House number in address2*: Shows the home number in a separate field.

Bulk operations

Here you can choose to

- *Add all products* - Marks all products for the Beslist feed
- *Add standard categories* - Put the Beslist.nl categories in a best effort way on the products

- *Mark 1000 as updated* - Marks 1000 products so that they can be reported via the Shopitem API.

Categories

The Beslist categories are saved in your PrestaShop installation, so that you can select the category for each product. You can retrieve the categories from Beslist by setting *Update Beslist categories* to YES', and then clicking on save.

- 'Update Beslist categories': Refresh the local list of categories of Beslist
- *Default category*: Select a default category for your products. For more information about categories you can look at the :doc:'categories page <usage/categories>' _

Live connection

After you have configured the test connection and performed the tests correctly, you can configure the module for production mode. For that only a few fields need to be changed:

- 'Use test connection': Set this field to 'NO'
- *Shopitem API key*: Your Shopitem API key

You are now ready to receive orders. Read in the :doc:'user documentation <usage/orders>' _ how this works.

1.3 Auto import

On the module configuration page you can click on 'view advanced options', there is a link to a script that performs a synchronization.

The link contains a secret key, so that only you can call the synchronization. The url looks like:

```
http://www.yourdomainname.com/modules/beslistcart/cron.php?secure_key=YOUR_SECURE_KEY
```

It is highly recommended to set up a cron task that calls this script every 15 minutes. Running the script is a light task, and can therefore often happen. You can create a cron task such as:

```
*/10 * * * * curl --silent http://www.yourdomainname.com/modules/beslistcart/cron.php?
↪secure_key=YOUR_SECURE_KEY &> /dev/null
```

This task retrieves new orders from Beslist.nl every 10 minutes.

1.4 Products

After installing the module, new options have appeared in the Prestashop menu. One of them is *Catalog* -> 'Beslist.nl products'. When you navigate there, you will see an overview of all products in the system that are published on Beslist.nl.

1.4.1 Selecting products

Within Prestashop you can now set per product how it should be displayed on Beslist.nl. On the product editing page (under *Catalog*) you will find a tab with the title 'Beslist.nl Shopping cart integration'. On that tab are the settings for Beslist.nl.

Catalogus / Beslist.nl products

Beslist.nl products

 Synchroniseer producten
  Help





BESLIST.NL PRODUCTS 18689						   	
Beslist Product ID ▼▲	Product ▼▲	Productcombinatie ▼▲	Gepubliceerd ▼▲	Gesynchroniseerd ▼▲	Winkel		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	--	<input type="text"/>	<input type="button" value="Zoeken"/>
1	Testproduct 1	0	✓	OK	Winkel	<input type="button" value="Bekijken"/>	
2	Testproduct 2	0	✓	OK	Winkel	<input type="button" value="Bekijken"/>	
3	Testproduct 3	0	✓	OK	Winkel	<input type="button" value="Bekijken"/>	

Fig. 1: Beslist.nl producten

On this tab you will see an option for all product combinations to publish them (via the tick under 'Publish '). You can also set a specific Beslist Delivery Time per combination.

When you click on the title of a product, you can adjust the delivery times:

- *Custom Delivery time NL*: an override for Beslist.nl
- *Custom Delivery time BE*: an override for Beslist.be

1.4.2 Update products

The module ensures that a message is sent to Beslist.nl after certain adjustments have been made (currently these are adjustments to stock and price). This report happens under water, so you normally do not notice it.

1.4.3 Synchronize products


When a new product is selected for Beslist, it is not included until after the next product feed run. So it usually takes a while before your product is visible. After the product is visible, stock and price updates can be processed quickly (via the normal update).

In the Beslist.nl product overview in your Prestashop installation you will see a list of all your products that have been selected for Beslist. A status is shown here. The following statuses are possible for a Decided publication:

- *OK* - This product has been correctly reported to Determined
- *New* - This product is new and has not yet (properly) been reported to Beslist
- *Stock updated* - The stock of this product has been updated, but has not (properly) been reported
- *Info updated* - The information (eg the price) of this product has been updated, but has not (properly) been reported

If there is a status other than 'OK' in the product overview, the 'Synchronize products' button at the top of the synchronization can be started manually. If you receive error messages during synchronization, you will receive a message. If the API is unavailable, you can try again later, but if incorrect data is sent, you may need to adjust the products.

BESLIST.NL SETTINGS



This interface allows you to edit the Beslist.nl shopping cart data.
You can also specify product/product combinations.


Beslist category ID

For a complete overview, look at: <https://www.beslist.nl/categories/>


Beslist category

BESLIST.NL PRODUCTS


Published	Product
<input type="checkbox"/>	-- All products --
<input checked="" type="checkbox"/>	Product x
	<div>Custom Delivery time NL (optional)</div> <input type="text"/>
	<div>Custom Delivery time BE (optional)</div> <input type="text"/>



Annuleren



Save and stay



Opslaan

Fig. 2: Beslist.nl producttab

1.4.4 Frequently Asked Questions

Why do I get 404 notifications when I save my product?

This is caused by the fact that the product is not yet known by Beslist. Certainly the product feed must first be re-read. This happens according to a certain schedule (at the time of writing 1 time a day at 8:00). For more information, see the [beslist handleidingen](#).

The products with 404 notifications can be offered the next day via the *Synchronize products* button. Then it continues to go well.

1.5 Productfeed

The module has a possibility to generate a feed. The generator must be set via cron. An example of a command can be found on the module configuration page (under 'View advanced options').

For instance:

```
0 1 * * * curl -L --max-redirs -1 https://www.uwdomeinnaam.nl/modules/beslistcart/  
↪cron-generate.php?secure_key=XXXYYY111222333 &>/dev/null
```

This task will run at 1:00 every night. Depending on the number of products, the task may take a long time. You can also use an external feed. It is known that the module works well with [Channable](#). You have more control over your feed via an external party.

1.6 Categories

From version 1.3.0 there is more extensive support for the categorization of Beslist. In addition to your own categorization, it is possible to send a predefined Beslist mapping with the products. This can be done on 3 levels: shop, shop category and product.

1.6.1 Beslist internal mapping

The default category of each product is included in the feed. On the basis of this categorization, Beslist creates a mapping to their own categories. So in principle you do not have to do anything to get the products in the right categories.

1.6.2 Your own mapping

**** The steps below are not necessary for most cases, in consultation with Beslist will see if you can create your mapping yourself ****

Categorization at shop level

You can specify on the module configuration page which default category to use for Beslist. This will be included if no other match is possible.

Categorization on shop category

In your Prestashop environment you can go to *Catalog -> Categories*. Select the category for which you want to set a Beslist category. On the edit page there is a field *Beslist category*. This category will be used when there is no match at product level.

The matching on shop category works through the entire tree of your shop. So if you have set a Beslist category at a higher level, it will be used.

Categorization at product level

If you can not indicate which category corresponds to the product on the shop and shop category level, you can indicate a category for the product on the product page itself. This category will always be included in the feed.

Determination method per product

In the feed a corresponding category is searched per product, this works as follows:

1. If a Beslist category is entered for the product, the feed uses it
2. If a Beslist category is entered for the (main) category of the product, the feed uses it
3. **If a Beslist category is found in the parent categories of the product, the feed uses it**
 - We look at which category is the deepest, so it may be that the main category of the product is not used, but another category where the product is also.
4. If there is still no match available, the default category of the shop is used in the feed

1.7 Orders

After installing the module, new options have appeared in the Prestashop menu. One is *Orders -> 'Beslist.nl orders'*. When you navigate there, you will see an overview of all Declined orders in the system (both from the Netherlands and from Belgium).

BESLIST.NL PRODUCTS 18689								
Beslist Product ID	Product	Productcombinatie	Beslist specifieke prijs	Gepubliceerd	Gesynchroniseerd	Winkel		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1	Testproduct 1	0	90,90 €	✓	OK	Winkel	<input type="button" value="Bekijken"/>	
2	Testproduct 2	0	18,30 €	✓	OK	Winkel	<input type="button" value="Bekijken"/>	
3	Testproduct 3	0	29,80 €	✓	OK	Winkel	<input type="button" value="Bekijken"/>	

Fig. 3: Beslist.nl orders

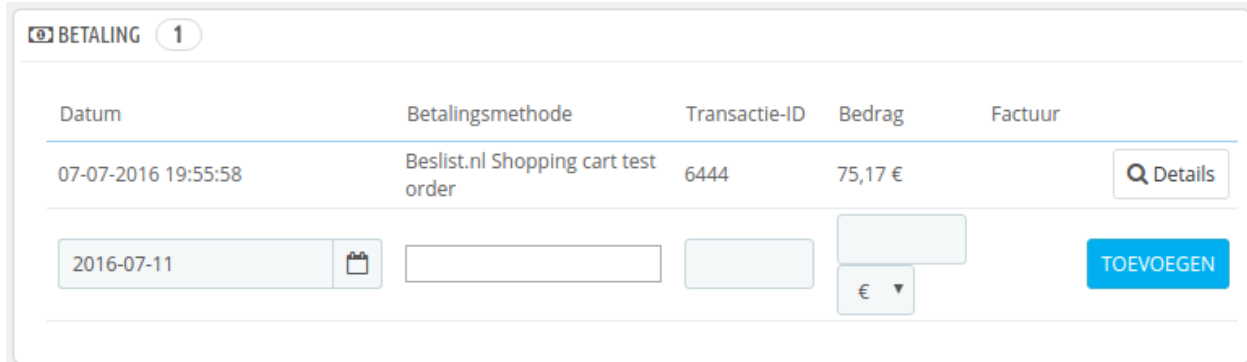
1.7.1 Orders overview

Navigate to *Orders -> 'Beslist.nl orders'*. You should see 'Synchronize orders' at the top of the screen.

With this button the orders of Beslist are retrieved and put in your system. You can then process the order through your internal order processing process. You can click on the order to navigate to the Prestashop order screen.

1.7.2 Prestashop order screen

In the Prestashop order screen you can find the Decide order reference in the 'Payment' panel.



The screenshot shows the 'BETALING' (Payment) panel in Prestashop. It contains a table with the following data:

Datum	Betalingsmethode	Transactie-ID	Bedrag	Factuur
07-07-2016 19:55:58	Beslist.nl Shopping cart test order	6444	75,17 €	Details

Below the table, there is a date selector set to '2016-07-11', a calendar icon, an empty input field, another empty input field, a currency dropdown set to '€', and a blue button labeled 'TOEVOEGEN' (Add).

Fig. 4: Order reference

1.7.3 Possible errors

When synchronizing, it is possible that an error is given. Here you will find an overview.

Can not find a product

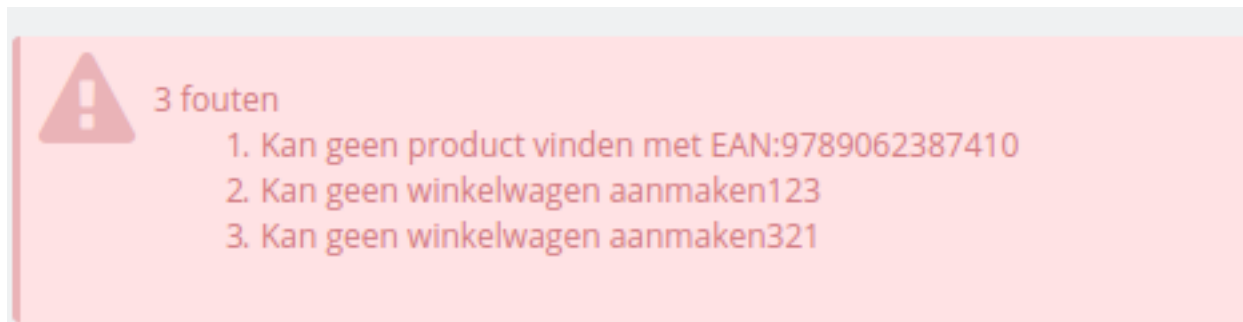


Fig. 5: Product not known

The error message *Can not find a product with eg code: XXXXX* is caused by the fact that there is no product with the number that has been reported back by Beslist. Create a product with this number and synchronize the orders again. The other errors (*Can not create a shopping cartXXX*) are caused by the first error. Check for certainty also what value you have entered in the *Decision product reference field* configuration of the module.

Warning when importing multiple orders

The warning is due to a bug in Prestashop core, the problem can not hurt, but a [bug report] (<http://forge.prestashop.com/browse/PSCSX-7858>) has been created. You can remove the warning by manually creating an override class for the Cart class (see code block below).

```

/**
 * Get the delivery option selected, or if no delivery option was selected,
 * the cheapest option for each address
 *
 * @param Country|null $default_country
 * @param bool         $dontAutoSelectOptions
 * @param bool         $use_cache
 *
 * @return array|bool|mixed Delivery option
 */
public function getDeliveryOption($default_country = null, $dontAutoSelectOptions = 
↪false, $use_cache = true)
{
    static $cache = array();
    $cache_id = (int)(is_object($default_country) ? $default_country->id : 0).'-'.
↪(int)$dontAutoSelectOptions.'-'. $this->id;
    if (isset($cache[$cache_id]) && $use_cache) {
        return $cache[$cache_id];
    }

    $delivery_option_list = $this->getDeliveryOptionList($default_country);

    // The delivery option was selected
    if (isset($this->delivery_option) && $this->delivery_option != '') {
        $delivery_option = Tools::unSerialize($this->delivery_option);
        $validated = true;
        foreach ($delivery_option as $id_address => $key) {
            if (!isset($delivery_option_list[$id_address][$key])) {
                $validated = false;
                break;
            }
        }

        if ($validated) {
            $cache[$cache_id] = $delivery_option;
            return $delivery_option;
        }
    }

    if ($dontAutoSelectOptions) {
        return false;
    }

    // No delivery option selected or delivery option selected is not valid, get the 
↪better for all options
    $delivery_option = array();
    foreach ($delivery_option_list as $id_address => $options) {
        foreach ($options as $key => $option) {
            if (Configuration::get('PS_CARRIER_DEFAULT') == -1 && $option['is_best_
↪price']) {
                $delivery_option[$id_address] = $key;
                break;
            } elseif (Configuration::get('PS_CARRIER_DEFAULT') == -2 && $option['is_
↪best_grade']) {
                $delivery_option[$id_address] = $key;
                break;
            } elseif ($option['unique_carrier'] && in_array(Configuration::get('PS_
↪CARRIER_DEFAULT'), array_keys($option['carrier_list']))) {

```

(continues on next page)

(continued from previous page)

```

        $delivery_option[$id_address] = $key;
        break;
    }
}

reset($options);
if (!isset($delivery_option[$id_address])) {
    $delivery_option[$id_address] = key($options);
}

$cache[$cache_id] = $delivery_option;

return $delivery_option;
}

```

I can not import products that are not in stock ^^^

This is possible, but then the products must be orderable (including on your website). You can change this functionality by creating an override class for Product, and putting the following code in it (see opposite / below)

```

public static function isAvailableWhenOutOfStock($out_of_stock)
{
    // @TODO 1.5.0 Update of STOCK_MANAGEMENT & ORDER_OUT_OF_STOCK
    static $ps_stock_management = null;
    if ($ps_stock_management === null) {
        $ps_stock_management = Configuration::get('PS_STOCK_MANAGEMENT');
    }

    if (!$ps_stock_management || isset(Context::getContext()->employee)) {
        return true;
    } else {
        static $ps_order_out_of_stock = null;
        if ($ps_order_out_of_stock === null) {
            $ps_order_out_of_stock = Configuration::get('PS_ORDER_OUT_OF_STOCK');
        }

        return (int)$out_of_stock == 2 ? (int)$ps_order_out_of_stock : (int)$out_of_
↪stock;
    }
}

```



channable

1.8 Channable integration

It is possible to use the module in combination with a 3rd party such as Channable. For this it is necessary that the layout of the ID is set with Channable. If you use the standard setting, it is possible that it is not yet right. The settings

below ensure that you can work with the *Standard* productmatcher setting of the module.

1.8.1 Product IDs

The feed must specify a product ID that matches the product ID that is returned by the Order API. Within Channable this field is known as the *store product code*. This field is set by default to the Prestashop database ID of the product.

Check whether this is actually the case. You can do this on the tab ‘finalize’ of your Beslist Shopping cart channel. Here the field *store product code* must be set to ‘id’.

The screenshot shows a configuration field for 'winkelproductcode'. On the left, there is a dropdown menu with 'id' selected. To the right of the dropdown is a blue button labeled 'winkelproductcode'. Further right is a question mark icon and the word 'Mandatory'.

Fig. 6: winkelproductcode should be on id

Also check on the ‘rules’ tab if there are no rules that modify the id. The only adjustment to the ‘id’ field is the adjustment described below.

1.8.2 Productcombination IDs

If your webshop uses product combinations (products with different sizes / colors) you have to add a line to your Beslist Shopping Cart Channel, so that the combination is sent instead of the product.

If your products use combinations, you have a variant field on the product screen of Channable (see image below). If that is not the case, you can contact Channable to have this changed.

upc		
	combination_id	1973
	ean	
variant	quantity	0
	reference	Test
	variant_price	0.000000

Fig. 7: variant field

If you have the variant field, you can create a rule so that the *id* is adjusted when the product is a combination. You do this by creating a new line at the *Rules* tab of your Beslist Shopping cart channel. The rule must have the following settings:

- *Name*: Prestashop variant ID
- *If*: variant.combination_id isn’t empty
- *Then*: take *id* and *combine value*: variant.combination_id-id (Click on the plus and click in the blue box to select a field)

If it is good, the rule will look like this:

The screenshot shows the 'Name' field set to 'Prestashop variant ID'. Below this, the 'If' condition is configured with 'variant.combination_id' and 'isn't empty'. The 'Then' action is set to 'take' the 'id' field and 'combine value'. A visual formula bar below the 'Then' section shows 'variant.combinati' (truncated) minus 'id', with a plus sign button to add more fields. A 'Save rule' button is at the bottom left.

Fig. 8: variant rule

1.8.3 Test

You can now look at the product in your feed. The product for which the combination exists should now have field like

```
<winkelproductcode>id_combinatie-id_product</winkelproductcode>
```

Also for the products without combinations the field should look like

```
<winkelproductcode>id_product</winkelproductcode>
```