



# PrestaShop

## <PRESTASHOP/>

*A practical userguide to integrate Prestashop ecommerce and Stanley/Stella API*

## Prerequisites

This guide will explain how to connect a Prestashop 1.7 webshop with Stanley/Stella API Framework to periodically synchronize the product referential.

The guide assumes that you are familiar with Prestashop and that you already have your Prestashop running.

It also requires a little bit of development knowledge in order to transform the CSV files.

In this guide, we will illustrate the integration in the PHP development environment but this could also be realized in any other modern development environment.

Please note that the PHP code provided in this document is for illustration purpose only.

Further technical explanation on the API can be found at <https://api.stanleystella.com>

The demo website built with these instructions can be found at <https://testps.stanleystella.com/en/>

## Table of Contents

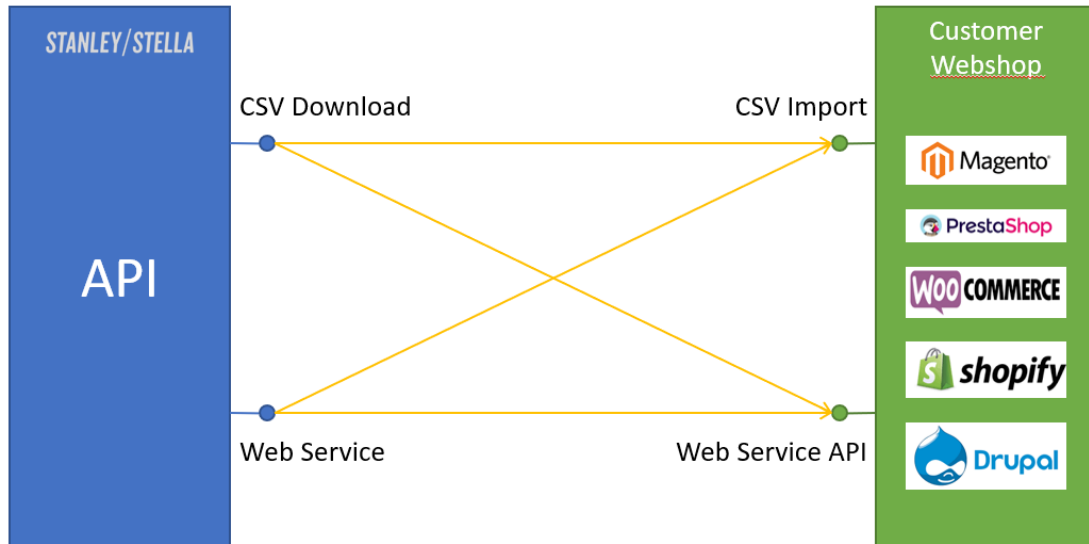
Prerequisites .....	2
Integration scenario Stanley/Stella API → Prestashop .....	4
Connect to Stanley/Stella Product API in PHP .....	4
Import CSV's in Prestashop.....	5
Brief description of the Prestashop Import CSV's .....	5
How to import CSV's in Prestashop .....	7
Generating the right CSV import files .....	8
Creating the “categories” .....	8
Generating the Product CSV import file .....	9
Importing the Product CSV .....	9
Generating the Combination CSV import file .....	10
Next step: Automation.....	10
PHP Code.....	11
Product CSV generation .....	11
Combination CSV generation .....	15

## Integration scenario Stanley/Stella API → Prestashop

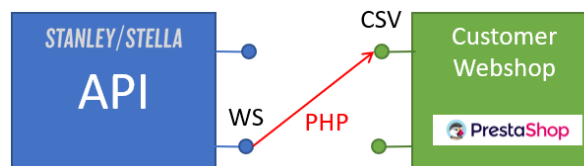
Like a lot of ecommerce solutions, Prestashop provides 2 ways to import product information:

- Manual import via CSV file (can be scheduled)
- Programmatic import via REST API

At Stanley/Stella, our API provides these 2 capabilities as well.



In this scenario, we chose to use the REST API on Stanley/Stella side and the CSV import on Prestashop side.



The integration logic in the middle has been developed in PHP.

## Connect to Stanley/Stella Product API in PHP

The first step of the integration is to call the Stanley/Stella Product REST API from the PHP code.

We use the Products V2 API which already groups the variants per style.

For simplicity, we only download the product data in 1 language (English).

See the following PHP code as inspiration.

```
function getAllSTSTProductsV2($language) {  
    $url = "https://api.stanleystella.com/webrequest/productsV2/get_json";  
    $jsonData = array(  
        'jsonrpc' => '2.0',  
        'method' => 'call',  
        'params' => array(  
            'db_name' => "production_api",  
            'password' => "YOUR_PASSWORD",  
            'user' => "YOUR_LOGIN",
```

```

        "LanguageCode" => $language,
    ),
    'id' => 0
);

$ch = curl_init($url);
$jsonDataEncoded = json_encode($jsonData);

curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
$result = curl_exec($ch);

$jsonDataDecoded = json_decode(json_decode($result)->result, true);
curl_close($ch);

return $jsonDataDecoded;
}

```

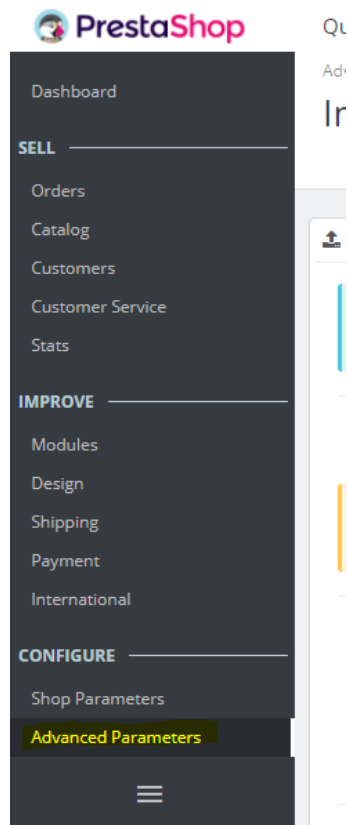
## Import CSV's in Prestashop

### Brief description of the Prestashop Import CSV's

Prestashop offers a CSV import functionality to import data like categories, products, variants (called combinations in Prestashop), customers, etc.

You can find more information about this functionality in the Prestashop documentation, f.ex <http://doc.prestashop.com/display/PS17/Import>.

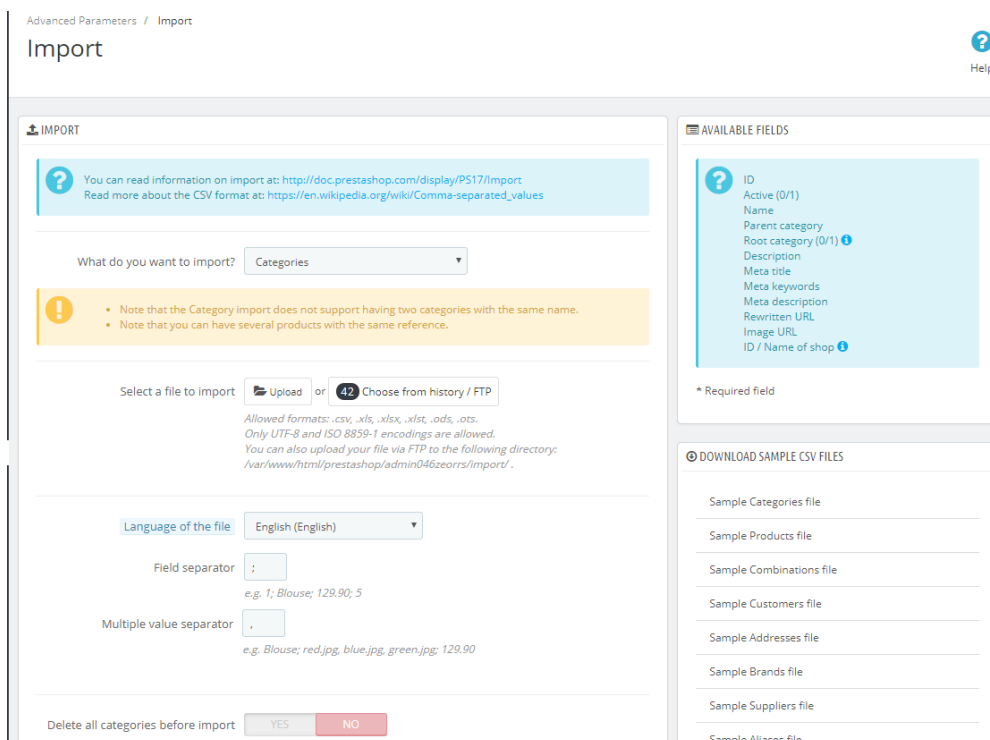
The import functionality can be found under “Advanced Parameters” and Import.



We will mainly use 3 CSV files, in this order:

- To import the categories
- To import the products
- To import the combinations (variants)

It is possible to download sample CSV files for each of these objects.



Unfortunately, the CSV import files of Prestashop require using the Prestashop internal id's. Therefore, we will have to use the Prestashop REST API's to query Prestashop and retrieve the internal id's.

The good news is that Prestashop takes care of identifying during import if it is a creation or an update based on the internal id's.

## How to import CSV's in Prestashop

When importing CSV's in Prestashop, it is important to select the right settings.

Advanced Parameters / Import

### Import

What do you want to import?

**!** Note that the Category import does not support having two categories with the same name.  
• Note that you can have several products with the same reference.

Select a file to import  or

Allowed formats: .csv, .xls, .xlsx, .xlwt, .ods, .ots.  
Only UTF-8 and ISO 8859-1 encodings are allowed.  
You can also upload your file via FTP to the following directory: /var/www/html/prestashop/admin046zeors/import/.

Language of the file:

Field separator:   
e.g. 1; Blouse; 129.90; 5

Multiple value separator:   
e.g. Blouse; red.jpg, blue.jpg, green.jpg; 129.90

Delete all products before import:

Use product reference as key:

Skip thumbnails regeneration:

Force all ID numbers:

Send notification email:

- Select the right object you want to import (f.ex Products)
- Select the corresponding CSV file
- Select the language you will import (each language must be separately imported)
- Be sure to define the right column separator of your CSV. It can be a comma ',' or a semi-colon ';'. In our case, we use the semi-colon.
- Be sure to define the right value separator of your CSV. In our case, we use the comma.
- In the selectors, there are 2 important selectors to be set:
  - o Use product reference as key (This will only be used if the product reference is an integer, which is not the case at Stanley/Stella) : YES
  - o Force all ID numbers : YES

If everything is correct, click 'Next'.

In the next screen, Prestashop shows you the mapping between your CSV columns and the data in Prestashop. Our example PHP programs generate the CSV in the standard import format for Prestashop 1.7.

**MATCH YOUR DATA**

Please match each column of your source file to one of the destination columns.

Save your data matching configuration  Save

Rows to skip

Indicate how many of the first rows of your file should be skipped when importing the data. For instance set it to 1 if the first row of your file contains headers.

ID	Active (0/1)	Name	Categories (x,y,z...)	Price tax excluded	Tax rule ID
Product ID	Active (0/1)	Name *	Categories (x,y,z...)	Price tax included	Tax rules ID
51	1	Stella Traces	26	19	1
52	1	Stella Tenders	22	19.95	1
53	1	Stella Plays	23	10	1
54	1	Stella Delights	21	12.35	1
55	1	Tell	28	43.7	1
56	1	Base	28	20.8	1
57	1	Stella Reduces	22	20.9	1
58	1	Stanley Adores	15	4.95	1
59	1	Live	27	4.35	1

Cancel Import

Be sure to fill in “1” in the field “Rows to skip” since we use the first line of the CSV for the headers of the columns.

If everything is correct, click “Import”.

## Generating the right CSV import files

### Creating the “categories”

You can either create the categories manually or import these categories via CSV.

We created the categories manually and noted then the internal id’s. We used these internal id’s in the product CSV import file.

CATEGORIES (4)

ID	Name	Description	Position	Displayed	
8	Women	For Her.	1	✓	View
12	Men	For Him.	2	✓	View
13	Unisex	For Both.	3	✓	View
14	Kids	For Kids.	4	✓	View

Bulk actions



CATEGORIES > WOMEN 6

ID	Name	Description	Position	Displayed	
--				--	Search
21	Tee-shirts		1	✓	Edit
22	Sweatshirts		2	✓	Edit
23	Polos		3	✓	Edit
24	Jackets		4	✓	Edit
25	Jogging shorts		5	✓	Edit
26	Pants		6	✓	Edit

Bulk actions

## Generating the Product CSV import file

The Product CSV import file has 1 line per style. So, we first call the Stanley/Stella Product V2 API, with variants grouped per style. We then loop on all styles.

For each style:

- We first try to retrieve the internal id of the product in Prestashop. To do this, we use the REST API of Prestashop : `/products/?filter[reference]=`
- If the result is empty, it means it is a new product. If the result is not empty, we will fill in this id in the CSV and Prestashop will take care of updating the product.
- We map the "Gender" and "Category" to the right Prestashop category and get the corresponding internal id.
  - o We add the style in the global collection. (All styles)
  - o We add Boys and Girls genders under Kids. The other genders stay as they are.
  - o For the other genders, we add the style under the right category.
  - o We also add the unisex styles under both Men and Women.
  - o Finally, we use the "NewStyle" flag of the Products API to add the new products under the "New" category.
- We replace `\n` characters in the long description by `<br>` HTML codes.
- In Prestashop, the price is first defined at the product level. Variations on the price per variants are then defined as differences with the product price (see generating the combination CSV import file). What we did is to take the price of the first variant and use this as the product price.
- We use the "StylePublished" flag to determine the "active" flag in Prestashop.
- We add the fit, neckline, sleeve as tags.
- We use the "MainPicture" data from the Products API to mention in the CSV the URL to the main picture. Prestashop takes care of downloading this picture and importing it.
- We then add a row in the CSV with all data at style level.

See the PHP code in the annexes.

## Importing the Product CSV

We first need to import the products in PHP. When generating the combinations CSV, we will retrieve the corresponding internal id's of the styles.

## Generating the Combination CSV import file

The Combination CSV import file has 1 line per variant. So, again, we first call the Stanley/Stella Product API, with variants grouped by style. We then loop on all styles and all variants.

The Combination CSV file does not have the issue with the id's of the categories. But it requires the internal id of the style to which the combination must be attached. Just like in the generation of the products csv, we retrieve the internal id's by using the REST API of Prestashop.

In Prestashop, the price of the variants is define as a delta from the price of the product. As described in the product CSV generation, we use the price of the first variant as base price.

During the generation of the combination CSV file, we also take care of the pictures. Therefore, for each style, we call the Stanley/Stella Product Image API to retrieve the URL's of the related pictures. We sort them by color code.

Note that there is a bug in Prestashop (or is it a feature?). When we import the products (at style level), we define the cover image and it is imported in Prestashop. When we import the combinations, with different pictures per colour, Prestashop displays as cover picture the first picture of the first variant, even if the picture selected for the cover is still the one from the product. As a workaround, we define first the variants of the main colour (that we retrieve from the main picture) and we import the picture in a specific order.

For each style and each variant:

- We calculate the difference between the price selected for the product (price of the first variant) and the price of the current variant.
- We group the variants by color to easily find the variants of the main colour.
- We first add the rows in the CSV for the main colour.
- We then add the rows in the CSV for the other colours.
- We concatenate the URL's of the pictures, separated by a comma, and fill this in the CSV line. We concatenate the picture in the right order : SFM0, SBM0, PFM0, PBM0, SFM5, SBM5

## Next step: Automation

With this CSV import technique and the need to use internal id's, it is difficult to fully automate the product import process.

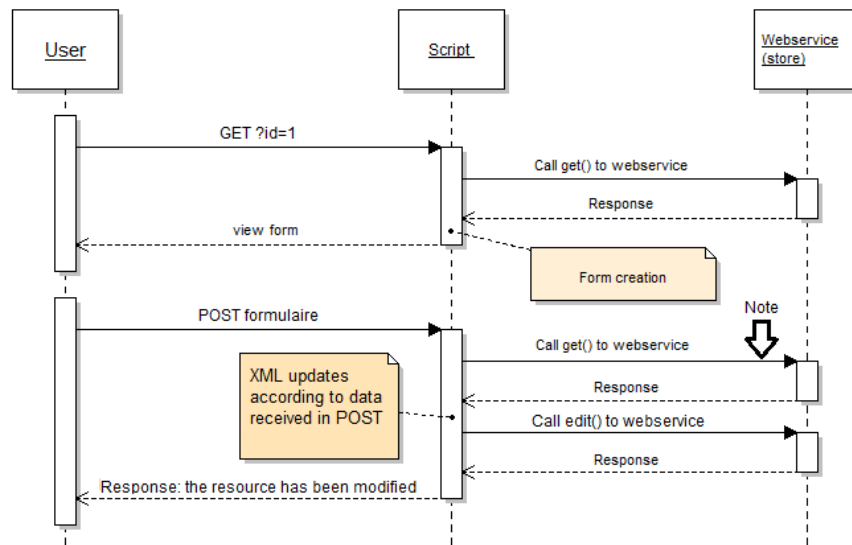
In order to fully automate this process, we suggest using the Prestashop REST API, instead of importing with CSV files, not only to ask Prestashop for the right internal id's for categories and for existing styles but also to create or update the categories, products and combinations.

Documentation of the Prestashop REST API can be found here:

- <http://doc.prestashop.com/display/PS16/Web+service+tutorial>
- <http://doc.prestashop.com/display/PS16/Web+service+reference>

See reference flow for update of objects via the API, according to the Prestashop documentation.

Sequence diagram representing how a resource is updated:



## PHP Code

See the following PHP code as inspiration for generating the CSV files.

### Product CSV generation

```

<?php
ini_set("memory_limit","128M");
ini_set("max_execution_time",9000);
ini_set('default_charset','utf-8');
header('Content-Type: text');

$domain = 'YOUR_PRESTASHOP_DOMAIN';
$key = 'YOUR_PRESTASHOP_API_KEY';
$baseurl = "https://".$key."@".$domain."/api";

// Prestashop categories
// Replace by your internal id's or use the Prestashop REST API
$categories = array();
$categories["All"] = 40;
$categories["Women"] = array();
$categories["Women"]["id"] = 3;
$categories["Women"]["TEES"] = 21;
$categories["Women"]["SHIRT"] = 32;
$categories["Women"]["SWEATS"] = 22;
$categories["Women"]["DRESSES"] = 33;
$categories["Women"]["POLO"] = 23;
$categories["Women"]["JACKETS"] = 24;
$categories["Women"]["PANTS"] = 26;
$categories["Men"] = array();
$categories["Men"]["id"] = 12;
$categories["Men"]["TEES"] = 15;
$categories["Men"]["SHIRT"] = 34;
$categories["Men"]["SWEATS"] = 16;
$categories["Men"]["POLO"] = 17;
$categories["Men"]["JACKETS"] = 18;
$categories["Men"]["PANTS"] = 20;
$categories["Unisex"] = array();
$categories["Unisex"]["id"] = 13;
  
```

```

$categories["Unisex"]["TEES"] = 27;
$categories["Unisex"]["SWEATS"] = 28;
$categories["Unisex"]["JACKETS"] = 35;
$categories["Unisex"]["ACCESSORY"] = 36;
$categories["Unisex"]["SHOES"] = 36;
$categories["Kids"] = array();
$categories["Kids"]["id"] = 14;
$categories["NEW"] = 39;

$languagecode = 'en_US';
$stpm = getAllSTSTProductsV2($languagecode);
//print_r($stpm);

$csvlines = array();
// CSV Header
$csvlines[] = addPrestashopCSVHeader();

foreach ($stpm as $k => $style) {
    $stylecode = $style["StyleCode"];
    if ($style["StylePublished"]) {
        // Get id of product from Prestashop if already exists
        $prestaId = "";
        $p = getProductByReference($baseurl, $stylecode);
        if (isset($p->product->id)) {
            echo "found existing product in PS for $stylecode with id=".$p->product->id;
            $prestaId = $p->product->id;
        } else {
            echo "creating new in PS for $stylecode";
        }
    }

    $csvrow = array();
    $stylename = $style["StyleName"];
    // Categories
    $gender = $style["Gender"];
    $category = $style["CategoryCode"];
    $type = $style["Type"];
    // Add collection category id
    $categorystring = $categories["All"];
    // Add gender category id
    if ($gender=="Boys" || $gender=="Girls") {
        $gender = "Kids";
    }
    $categorystring.=",";
    $categorystring.= $categories[$gender]["id"];
    // Add type category id for all but kids
    if ($gender!="Kids") {
        $categorystring.=",";
        $categorystring.= $categories[$gender][$category];
    }
    // Add Men + Women if unisex
    if ($gender=="Unisex") {
        $categorystring.=",";
        $categorystring.= $categories["Men"]["id"];
        $categorystring.=",";
        $categorystring.= $categories["Women"]["id"];
    }
    // Add newstyle category id
    if ($style["Variants"][0]["NewStyle"]) {
        $categorystring.=",";
        $categorystring.= $categories["NEW"];
    }
    // Descriptions
    $shortdescription = $style["ShortDescription"];
    $longdescription = str_replace("\n", "<br/>", $style["LongDescription"]);
    // Tags
    $fit = $style["Fit"];
    $neckline = $style["Neckline"];
    $sleeve = $style["Sleeve"];
    // Price : take first variant price
    $firstvariant = reset($style["Variants"]);
    $price = $firstvariant["Price<10 EUR"];
    // Status
    if ($style["StylePublished"]) {
        $published = 1;
    }
}

```

```

    } else {
        $published = 0;
    }

    $csvrow[] = $prestaId;
    $csvrow[] = $published; // status
    $csvrow[] = $styleName;
    $csvrow[] = $categoryString;
    $csvrow[] = $price;
    $csvrow[] = 1;
    $csvrow[] = "";
    $csvrow[] = 0;
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = $styleCode;
    $csvrow[] = $styleCode;
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = 1;
    $csvrow[] = "both";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = $shortDescription;
    $csvrow[] = $longDescription;
    $tags = $category." ".$type." ".$gender." ".$fit." ".$neckline." ".$sleeve;
    $csvrow[] = $tags;
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = $published; // active
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = 1;
    if (isset($style["MainPicture"][0]["HTMLPath"])) {
        $imageurl = $style["MainPicture"][0]["HTMLPath"];
    } else {
        $imageurl = "";
    }
    $csvrow[] = $imageurl;
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = "";
    $csvrow[] = 0;
    $csvrow[] = "new";
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;
    $csvrow[] = 0;

    $csvlines[] = $csvrow;
}

}

// Write to CSV
$fp = fopen('prestashop_product_import2019.csv', 'w');
foreach ($csvlines as $key => $csvline) {
    fputcsv($fp, $csvline, ";");
}

```

```
fclose($fp);
```

```
function addPrestashopCSVHeader() {  
    $csvrow = array();  
    $csvrow[] = "Product ID";  
    $csvrow[] = "Active (0/1)";  
    $csvrow[] = "Name *";  
    $csvrow[] = "Categories (x,y,z...)";  
    $csvrow[] = "Price tax included";  
    $csvrow[] = "Tax rules ID";  
    $csvrow[] = "Wholesale price";  
    $csvrow[] = "On sale (0/1)";  
    $csvrow[] = "Discount amount";  
    $csvrow[] = "Discount percent";  
    $csvrow[] = "Discount from (yyyy-mm-dd)";  
    $csvrow[] = "Discount to (yyyy-mm-dd)";  
    $csvrow[] = "Reference #";  
    $csvrow[] = "Supplier reference #";  
    $csvrow[] = "Supplier";  
    $csvrow[] = "Manufacturer";  
    $csvrow[] = "EAN13";  
    $csvrow[] = "UPC";  
    $csvrow[] = "Ecotax";  
    $csvrow[] = "Width";  
    $csvrow[] = "Height";  
    $csvrow[] = "Depth";  
    $csvrow[] = "Weight";  
    $csvrow[] = "Quantity";  
    $csvrow[] = "Minimal quantity";  
    $csvrow[] = "Visibility";  
    $csvrow[] = "Additional shipping cost";  
    $csvrow[] = "Unity";  
    $csvrow[] = "Unit price";  
    $csvrow[] = "Short description";  
    $csvrow[] = "Description";  
    $csvrow[] = "Tags (x,y,z...)";  
    $csvrow[] = "Meta title";  
    $csvrow[] = "Meta keywords";  
    $csvrow[] = "Meta description";  
    $csvrow[] = "URL rewritten";  
    $csvrow[] = "Text when in stock";  
    $csvrow[] = "Text when backorder allowed";  
    $csvrow[] = "Available for order (0 = No, 1 = Yes)";  
    $csvrow[] = "Product available date";  
    $csvrow[] = "Product creation date";  
    $csvrow[] = "Show price (0 = No, 1 = Yes)";  
    $csvrow[] = "Image URLs (x,y,z...)";  
    $csvrow[] = "Image alt texts (x,y,z...)";  
    $csvrow[] = "Delete existing images (0 = No, 1 = Yes)";  
    $csvrow[] = "Feature (Name:Value:Position)";  
    $csvrow[] = "Available online only (0 = No, 1 = Yes)";  
    $csvrow[] = "Condition";  
    $csvrow[] = "Customizable (0 = No, 1 = Yes)";  
    $csvrow[] = "Uploadable files (0 = No, 1 = Yes)";  
    $csvrow[] = "Text fields (0 = No, 1 = Yes)";  
    $csvrow[] = "Out of stock";  
    $csvrow[] = "ID / Name of shop";  
    $csvrow[] = "Advanced stock management";  
    $csvrow[] = "Depends On Stock";  
    $csvrow[] = "Warehouse";  
    return $csvrow;  
}
```

```
function getAllSTSTProductsV2($language) {  
    $url = "https://api.stanleystella.com/webrequest/productsV2/get_json";  
    $jsonData = array(  
        'jsonrpc' => '2.0',  
        'method' => 'call',  
        'params' => array(  
            'db_name' => "production_api",  
            'password' => "YOUR_PASSWORD",  
            'user' => "YOUR_LOGIN",  
            "LanguageCode" => $language,  
        ),  
    ),
```

```

        'id' => 0
    );

    $ch = curl_init($url);
    $jsonDataEncoded = json_encode($jsonData);

    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    $result = curl_exec($ch);

    $jsonDataDecoded = json_decode(json_decode($result)->result, true);
    curl_close($ch);

    return $jsonDataDecoded;
}

function getProductByReference($baseurl, $stylecode) {
    $url = $baseurl."/products/?filter[reference]={$stylecode}";
    $response = file_get_contents($url);
    $xml = new SimpleXMLElement($response);
    if (!isset($xml->products->product)) {
        return null;
    }
    $url2 = $baseurl."/products/".$xml->products->product["id"];
    $response2 = file_get_contents($url2);
    $xml2 = new SimpleXMLElement($response2);
    return $xml2;
}

```

## Combination CSV generation

```

<?php
ini_set("memory_limit", "128M");
ini_set("max_execution_time", 9000);
ini_set('default_charset', 'utf-8');
header('Content-Type: text');

$domain = 'YOUR_PRESTASHOP_DOMAIN';
$key = 'YOUR_PRESTASHOP_API_KEY';
$baseurl = "https://".$key."@".$domain."/api";

$languagecode = 'en_US';
$stpm = getAllSTSTPProductsV2($languagecode);

$fp = fopen('prestashop_combination_import.csv', 'w');
// CSV Header
$csvrow = array();
$csvrow[] = "Product ID*";
$csvrow[] = "Product Reference";
$csvrow[] = "Attribute (Name:Type:Position)*";
$csvrow[] = "Value (Value:Position)*";
$csvrow[] = "Reference";
$csvrow[] = "Supplier reference";
$csvrow[] = "EAN13";
$csvrow[] = "UPC";
$csvrow[] = "Wholesale price";
$csvrow[] = "Impact on price";
$csvrow[] = "Ecotax";
$csvrow[] = "Quantity";
$csvrow[] = "Minimal quantity";
$csvrow[] = "Impact on weight";
$csvrow[] = "Default (0 = No, 1 = Yes)";
$csvrow[] = "Combination available date";
$csvrow[] = "Image position";

```

```

$csvrow[] = "Image URLs (x,y,z...)";
$csvrow[] = "Image alt texts (x,y,z...)";
$csvrow[] = "ID / Name of shop";
$csvrow[] = "Advanced Stock Managment";
$csvrow[] = "Depends on stock";
$csvrow[] = "Warehouse";
fputcsv($fp, $csvrow, ";");

```

```

foreach ($stpm as $skk => $style) {
    $stylecode = $style["StyleCode"];
    if ($style["StylePublished"]) {
        // Get id of product from Prestashop. Display error if product does not exist.
        $prestaaid = "";
        $p = getProductByReference($baseurl, $stylecode);
        if (isset($p->product->id)) {
            echo "Found existing product in PS for $stylecode with id=".$p->product->id;
            $prestaaid = $p->product->id;
            // Start with main color first
            $maincolor = $style["MainPicture"][0]["ColorCode"];
            // Price : take first variant price
            $firstvariant = reset($style["Variants"]);
            $firstprice = $firstvariant["Price<10 EUR"];
            // Get images for current style
            $picurls = array();
            $pics = getSTSTProductImages($stylecode);
            foreach ($pics as $key => $pic) {
                // Group by color
                $colorcode = $pic["ColorCode"];
                $stype = substr($pic["FName"],0,4);
                $imageurl = $pic["HTMLPath"];
                if (!isset($picurls[$stylecode])) $picurls[$stylecode] = array();
                if (!isset($picurls[$stylecode][$colorcode])) $picurls[$stylecode][$colorcode] =
array();

                $picurls[$stylecode][$colorcode][$stype] = $imageurl;
            }

            // Group variants by color
            $variants = array();
            foreach ($style["Variants"] as $k => $var) {
                if (!isset($variants[$var["ColorCode"]])) $variants[$var["ColorCode"]] = array();
                $variants[$var["ColorCode"]][] = $var;
            }

            // Output the main color
            $vars = $variants[$maincolor];
            foreach ($vars as $k => $var) {
                $colorname = $var["Color"];
                $colorcode = $var["ColorCode"];
                $sizename = $var["SizeCode"];
                $colorsequence = $var["ColorSequence"];
                $sizesequence = $var["SizeSequence"];

                $csvrow = array();
                $csvrow[] = $prestaaid;
                $csvrow[] = $var["B2BSKUREF"];

                $csvrow[] = "Color:color:". $colorsequence. ",Size:select:". $sizesequence;
                $csvrow[] = $colorname. ":". $colorsequence. ",". $sizename. ":". $sizesequence;

                $csvrow[] = $var["B2BSKUREF"];
                $csvrow[] = $var["B2BSKUREF"];
                $csvrow[] = "";
                $csvrow[] = "";
                $csvrow[] = "";
                $varprice = $var["Price<10 EUR"];
                $csvrow[] = $varprice - $firstprice;
                $csvrow[] = 0;
                $csvrow[] = $var["Stock"];
                $csvrow[] = 1;
                $csvrow[] = "";
                $csvrow[] = 0;
                $csvrow[] = $var["SKU_Start_Date"];
                $csvrow[] = "";
            }

```



```

// Define order : SFM0, SFM5, PFM0, PBM0, SBM0, SBM5
$allpicurls = "";
$sortedpics = array();
if (isset($picurls[$stylecode][$colorcode]["SFM0"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["SFM0"];
}
if (isset($picurls[$stylecode][$colorcode]["SFM5"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["SFM5"];
}
if (isset($picurls[$stylecode][$colorcode]["PFM0"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["PFM0"];
}
if (isset($picurls[$stylecode][$colorcode]["PBM0"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["PBM0"];
}
if (isset($picurls[$stylecode][$colorcode]["SBM0"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["SBM0"];
}
if (isset($picurls[$stylecode][$colorcode]["SBM5"])) {
    $sortedpics[] = $picurls[$stylecode][$colorcode]["SBM5"];
}
$first = true;
foreach ($sortedpics as $key => $picurl) {
    if (!$first) {
        $allpicurls.=",";
    }
    $allpicurls.=$picurl;
    $first = false;
}
$csvrow[] = $allpicurls;
$csvrow[] = "";

```

```

$csvrow[] = "";
$csvrow[] = "";
$csvrow[] = "";
$csvrow[] = "";

```

```

fputcsv($fp, $csvrow, ";");
}

```

```

// Output the other variants
foreach ($style["Variants"] as $l1 => $var) {
    $colorcode = $var["ColorCode"];
    if ($colorcode!=$maincolor) {
        $colorname = $var["Color"];
        $sizecode = $var["SizeCode"];
        $colorsequence = $var["ColorSequence"];
        $sizecode = $var["SizeSequence"];
    }
}

```

```

$csvrow = array();
$csvrow[] = $prestad;
$csvrow[] = $var["B2BSKUREF"];

```

```

$csvrow[] = "Color:color;".$colorsequence.",Size:select;".$sizecode;
$csvrow[] = $colorname.";".$colorsequence.",".$sizecode;

```

```

$csvrow[] = $var["B2BSKUREF"];
$csvrow[] = $var["B2BSKUREF"];
$csvrow[] = "";
$csvrow[] = "";
$csvrow[] = "";
$varprice = $var["Price<10 EUR"];
$csvrow[] = $varprice - $firstprice;
$csvrow[] = 0;
$csvrow[] = $var["Stock"];
$csvrow[] = 1;
$csvrow[] = "";
$csvrow[] = 0;
$csvrow[] = $var["SKU_Start_Date"];
$csvrow[] = "";

```

```

// Define order : SFM0, SFM5, PFM0, PBM0, SBM0, SBM5
// TODO : add pictures only on first size of color variant
$allpicurls = "";
$sortedpics = array();

```

```

        if (isset($picurls[$stylecode][$colorcode]["SFM0"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["SFM0"];
        }
        if (isset($picurls[$stylecode][$colorcode]["SFM5"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["SFM5"];
        }
        if (isset($picurls[$stylecode][$colorcode]["PFM0"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["PFM0"];
        }
        if (isset($picurls[$stylecode][$colorcode]["PBM0"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["PBM0"];
        }
        if (isset($picurls[$stylecode][$colorcode]["SBM0"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["SBM0"];
        }
        if (isset($picurls[$stylecode][$colorcode]["SBM5"])) {
            $sortedpics[] = $picurls[$stylecode][$colorcode]["SBM5"];
        }
        $first = true;
        foreach ($sortedpics as $key => $picurl) {
            if (!$first) {
                $allpicurls.=",";
            }
            $allpicurls.=$picurl;
            $first = false;
        }
        $csvrow[] = $allpicurls;
        $csvrow[] = "";

        $csvrow[] = "";
        $csvrow[] = "";
        $csvrow[] = "";
        $csvrow[] = "";

        fputcsv($fp, $csvrow, ",");
    }
} else {
    echo "$stylecode not found in Prestashop. Please create product first.";
}
}

fclose($fp);

function getAllSTSTProductsV2($language) {
    $url = "https://api.stanleystella.com/webrequest/productsV2/get_json";
    $jsonData = array(
        'jsonrpc' => '2.0',
        'method' => 'call',
        'params' => array(
            'db_name' => "production_api",
            'password' => "YOUR_PASSWORD",
            'user' => "YOUR_LOGIN",
            "LanguageCode" => $language,
        ),
        'id' => 0
    );

    $ch = curl_init($url);
    $jsonDataEncoded = json_encode($jsonData);

    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    $result = curl_exec($ch);

    $jsonDataDecoded = json_decode(json_decode($result)->result, true);
    curl_close($ch);

    return $jsonDataDecoded;
}

```

```
}  
  
function getProductByReference($baseurl, $stylecode) {  
    $url = $baseurl."/products/?filter[reference]=[$stylecode]";  
    $response = file_get_contents($url);  
    $xml = new SimpleXMLElement($response);  
    if (!isset($xml->products->product)) {  
        return null;  
    }  
    $url2 = $baseurl."/products/".$xml->products->product["id"];  
    $response2 = file_get_contents($url2);  
    $xml2 = new SimpleXMLElement($response2);  
    return $xml2;  
}
```