<SHOPIFY/>

A practical userguide to integrate Shopify ecommerce and Stanley/Stella API

Prerequisites

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

The guide assumes that you are familiar with Shopify and that you already started your shop on Shopify.

It also requires a little bit of internet development knowledge, specifically the use of REST API's.

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 build with these instructions can be found at https://ststdemo.myshopify.com

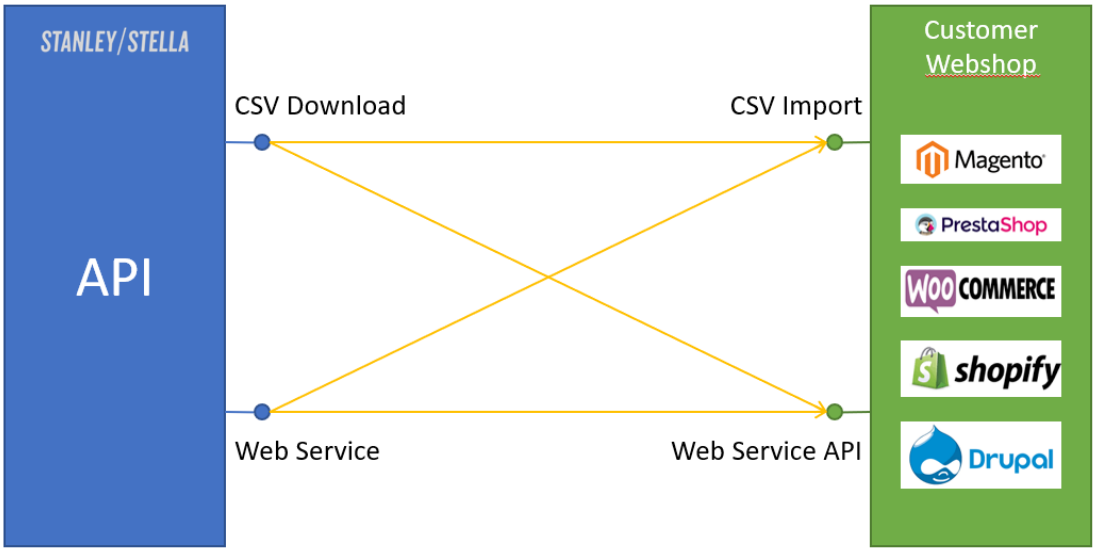
Table of Contents

Integration scenario Stanley/Stella API → Shopify

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

- Manual import via CSV file
- 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 both 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.

The standard output of the Stanley/Stella API is a “flat” list of product variants (SKU’s).

In our PHP code, we added the grouping of the variants at style level and the sorting of the variants by colors and sizes.

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

See the following PHP code as inspiration. Feel free to extend this code with additional data coming from the API.

```php
<?php

function getAllSTSTProducts() {
    $url = "https://api.stanleystella.com/webrequest/products/get_json";
```

```
    $jsonData = array(
        'jsonrpc' => '2.0',
        'method' => 'call',
        'params' => array(
            'db_name' => "production_api",
            'password' => "Test0220",
            'user' => "test@stanleystella.com",
            "LanguageCode" => "en_US"
        ),
        '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($ch);
    // Sort the data by style, color and size
    foreach ($jsonDataDecoded as $key => $row) {
        $styleorder[$key] = $row['SequenceStyle'];
        $sizeorder[$key]  = $row['SizeSequence'];
        $colororder[$key] = $row['ColorSequence'];
    }
    array_multisort($styleorder, SORT_ASC, $colororder, SORT_ASC, $sizeorder, SORT_ASC,
$jsonDataDecoded);

    return $jsonDataDecoded;
}

function getAllSTSTProductsGrouped() {
    $stpm = getAllSTSTProducts();

    $allststyles = array();
    foreach ($stpm as $key => $variant) {
        $stylecode = $variant["StyleCode"];
        if (!isset($allststyles[$stylecode])) {
            $allststyles[$stylecode] = array();
            $allststyles[$stylecode]["StyleCode"] = $variant["StyleCode"];
            $allststyles[$stylecode]["StyleName"] = $variant["StyleName"];
            $allststyles[$stylecode]["Type"] = $variant["Type"];
            $allststyles[$stylecode]["Category"] = $variant["Category"];
            $allststyles[$stylecode]["Gender"] = $variant["Gender"];
            $allststyles[$stylecode]["Fit"] = $variant["Fit"];
            $allststyles[$stylecode]["Neckline"] = $variant["Neckline"];
            $allststyles[$stylecode]["Sleeve"] = $variant["Sleeve"];
            $allststyles[$stylecode]["ShortDescription"] = $variant["ShortDescription"];
            $allststyles[$stylecode]["LongDescription"] = $variant["LongDescription"];
            $allststyles[$stylecode]["SequenceStyle"] = $variant["SequenceStyle"];
        }

        if (!isset($allststyles[$stylecode]["variants"])) $allststyles[$stylecode]["variants"]
= array();
        $var = array();
        $var["B2BSKUREF"] = $variant["B2BSKUREF"];
        $var["ColorCode"] = $variant["ColorCode"];
        $var["Color"] = $variant["Color"];
        $var["ColorSequence"] = $variant["ColorSequence"];
        $var["SizeCode"] = $variant["SizeCode"];
        $var["SizeCodeNavision"] = $variant["SizeCodeNavision"];
        $var["SizeSequence"] = $variant["SizeSequence"];
        $var["Stock"] = $variant["Stock"];
        $var["Price<10 EUR"] = $variant["Price<10 EUR"];
        $var["SKU_Start_Date"] = $variant["SKU_Start_Date"];
        $var["Published"] = $variant["Published"];
        $allststyles[$stylecode]["variants"][$variant["B2BSKUREF"]] = $var;

    }
```

```
    return($allststystyles);
}
```

Connecting to the Shopify REST API

Brief description of the Shopify REST API

The Shopify API is described on the developers platform of Shopify.

https://developers.shopify.com

The REST Admin API can be found under https://help.shopify.com/api/reference

The REST Admin API lets you access nearly all resources of Shopify (customers, products, orders, metafields, shipping and fulfilment, inventory, etc). In this scenario, we will use only the “product” functions of the API. This covers the following objects:

Collect	After creating a custom collection, add products to it by creating a collect for each product. Each collect associates one product with one custom collection.
CustomCollection	Manage a store's custom collections. A custom collection is one where products are included manually, as opposed to being included automatically because they meet selection conditions.
Product	Manage a store's products, which are the individual items and services for sale in the store.
Product Image	Add or update a store's product images, which sales channels use to display the products to shoppers.
Product Variant	Add or update a product's variants. Variants are the different combinations of the product's options. For example, a t-shirt product with size and color options might have a variant in a small size and blue color.
SmartCollection	Create or update smart collections by defining selection conditions. Products that match the conditions are included in the collection automatically.

The Shopify REST API uses the standard REST verbs (GET, PUT, POST, DELETE).

What you can do with Product

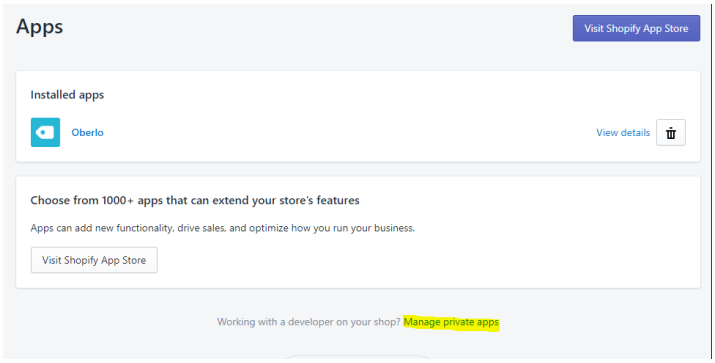
The Shopify API lets you do the following with the Product resource. More detailed versions of these general actions may be available:

- GET /admin/products.json
  Retrieves a list of products
- GET /admin/products/count.json
  Retrieves a count of products
- GET /admin/products/#{product_id}.json
  Retrieves a single product
- POST /admin/products.json
  Creates a new product
- PUT /admin/products/#{product_id}.json
  Updates a new product
- DELETE /admin/products/#{product_id}.json
  Deletes a product

Generating a Shopify API key

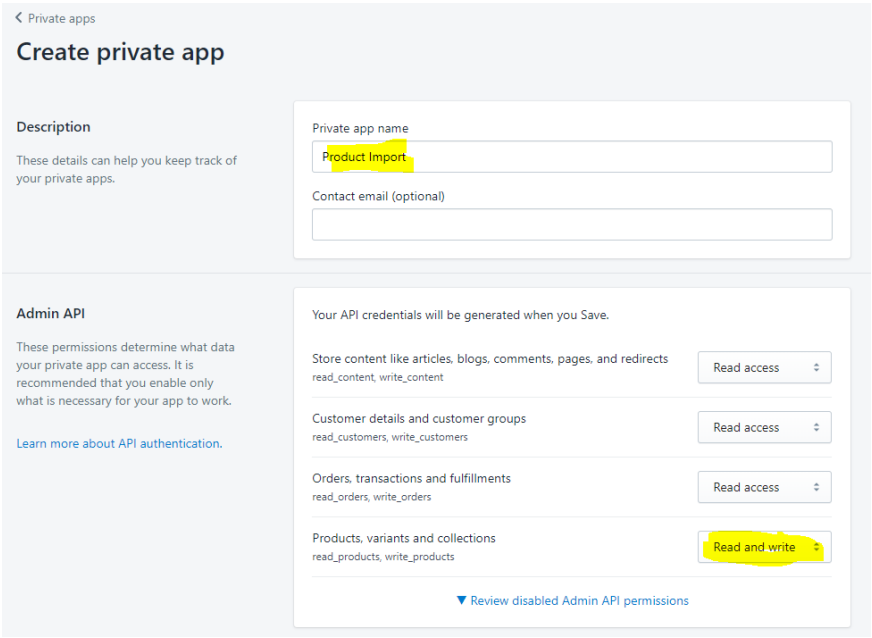
Before using the REST API, you must define an API key in Shopify.

You can do this by navigating to “Apps” and then click on “Manage private apps”.



Click then on “Create a new private app”.

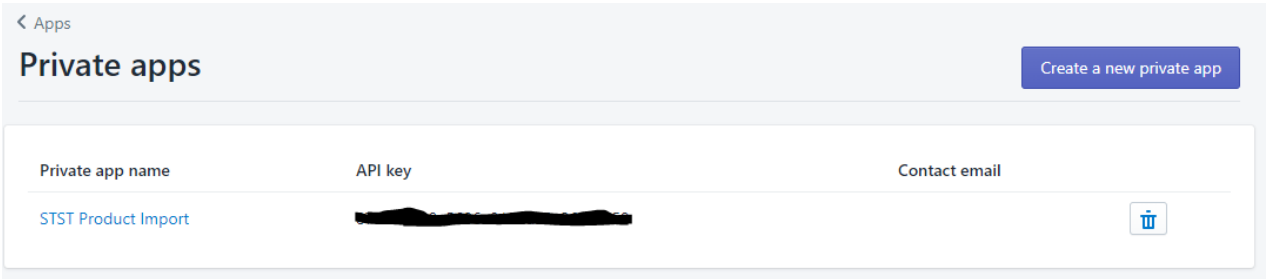
You need to give a name for your private app and then select the access rights.



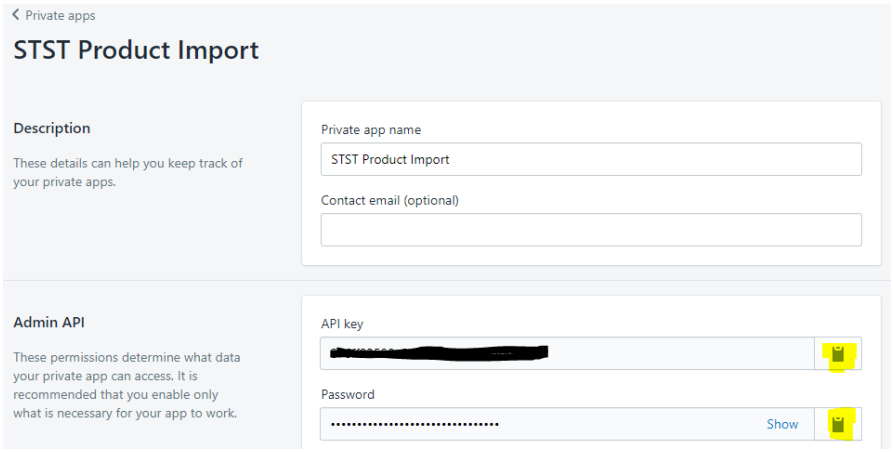
For products, variants and collections, select the “Read and write” access rights.

Click “Save”.

You should now see your app and the API key.



Click on your new app. We need to copy the API key and the associated password. Click on the “copy” icon to copy the API key and paste it in a temporary file editor (like Notepad). Do the same for the password.



PS: For security reasons, we erased our key from the screenshots.

The Shopify API will be called by using the following URL structure:

https://[YOUR-API-KEY]:[YOUR-API-PASSWORD]@[YOUR-SHOP-NAME.YOUR-DOMAIN]/admin/products.json

Calling the Shopify API from PHP

From PHP, we mainly call the Shopify API via GET or POST/PUT.

For the GET, we use the simple PHP method : file_get_contents($url);

For the POST, we use CURL and set the right options.

Make sure you enabled CURL in your PHP installation.

- Locate your PHP.ini file (normally located in the bin folder of your apache installation)
- Open the PHP.ini file in a text editor
- Search for : ;extension=php_curl
- Uncomment this line by removing the semi-colon ';'
- Save and close PHP.ini
- Restart Apache

See the following PHP code as inspiration. Change the key, password and domain to your settings.

```php
<?php
function getProductInShopifyByHandle($handle) {
    $url = "https://YOURKEY:YOURPASSWORD@YOURDOMAIN/admin/products.json?handle=".$handle;
    $response = file_get_contents($url);
    $jsonDataDecoded = json_decode($response, true);
    if (isset($jsonDataDecoded["products"][0])) return $jsonDataDecoded["products"][0];
    else return null;
}

function createProductInShopify($jsonData) {
    $url = "https://YOURKEY:YOURPASSWORD@YOURDOMAIN/admin/products.json";

    $jsonDataEncoded = json_encode($jsonData);

    $ch = curl_init($url);
    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);

    curl_close($ch);

    return $result;
}

function createProductImageInShopify($productid, $jsonData) {
    $url = "https://YOURKEY:YOURPASSWORD@YOURDOMAIN/admin/products/".$productid."/images.json";

    $jsonDataEncoded = json_encode($jsonData);

    $ch = curl_init($url);
    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);

    curl_close($ch);

    return $result;
}
```

Integration implementation

Shopify handle

We chose to import the Stanley/Stella products at style level. We used the style code as the unique handle for products in Shopify. Each product has then a set of variants for each color+size combination. (See Shopify limitations a little bit further in this document)

We also chose to insert most of the data as tags on the products. Shopify has then a nice way of re-using these tags for filters, navigation, menu's, etc.

Shopify categories and variants dimensions

A very handy functionality of Shopify is that categories and variant dimensions can be created on the fly during the product import via the API.

You don't need to pre-configure Shopify with the right categories, colors and sizes. This will all automatically be imported in Shopify.

Integration logic

The integration logic is then the following:

- Call the Stanley/Stella Product API, grouped by styles and sorted by colors and sizes
- For each style:
    - Ask Shopify if the style already exists. Use the stylecode as handle.

- If the style already exists, update it with a PUT call.
- If the style does not exist, create it with a POST call.
- We then create the Shopify PHP object. This contains:
  - Data at style level (like stylename, shortdescription, longdescription, type and tags for gender, fit, neckline, sleeves, etc)
  - For each variant:
    - The SKU reference
    - The colorcode and sizecode as option1 and option2 respectively
    - The price
    - The stock
    - Depending on your configuration, you will then have specific values for inventory management, inventory policy and fulfilment service. We chose:
      - Inventory management = shopify
      - Inventory policy = continue
      - Fulfillment service = manual
  - Shopify also request a final block of information containing the list of color and size values
- This object is then encoded in JSON and sent by POST to the REST API.

Importing the pictures

In the same loop, we also import the pictures.

To import pictures in Shopify, you just need to give the URL of the picture and Shopify will follow this link and download the picture internally. You also can link the picture to a specific variant in Shopify. Therefore, you first need to retrieve the internal Shopify id for the product and its variants. To do this, we ask Shopify with a REST GET to send us the data of the newly created product. We then group the variants by their colorcode.

We then just have to call the Stanley/Stella Product Image API to retrieve the pictures data for this particular style and assign the picture path to the right Shopify internal id for the corresponding variants.

This Shopify object is then encoded in JSON and sent by POST to the REST API.

Limitations of Shopify

Shopify allows only 100 variants per product. This is not enough to represent all colors and size combination for some of the Stanley/Stella products, available in many colors (like f.ex the Leads).

Shopify allows also only 300 pictures per product. Again, this can be too few for styles with a lot of colors and pictures.

PHP Code

See the following PHP code for inspiration of the integration logic.

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

$allststvariants = getAllSTSTProductsGrouped();

foreach ($allststvariants as $stylecode => $style) {
    // Check if style already imported. If yes : update else : create
    $shopifyproduct = getProductInShopifyByHandle($stylecode);
    if ($shopifyproduct==null) {
        // CREATE
        $allstyles = array();
        $stylename = $style["StyleName"];
        $shortdesc = $style["ShortDescription"];
        $longdesc = $style["LongDescription"];
        $type = $style["Type"];
        $category = $style["Category"];
        $gender = $style["Gender"];
        $fit = $style["Fit"];
        $neckline = $style["Neckline"];
        $sleeve = $style["Sleeve"];

        if (!isset($allstyles[$stylecode])) {  // Style level
            $allstyles[$stylecode] = new stdClass ();

            $allstyles[$stylecode]->product = new stdClass();
            $allstyles[$stylecode]->product->handle = $stylecode;
            $allstyles[$stylecode]->product->title = $stylename;
            $allstyles[$stylecode]->product->body_html = "<h3>".$shortdesc."</h3><p>".
str_replace("\n", "<br/>", $longdesc)."</p>";
            $allstyles[$stylecode]->product->vendor = "Stanley/Stella";
            $allstyles[$stylecode]->product->product_type = $type;
            $allstyles[$stylecode]->product->tags =
$category.",".$type.",".$gender.",".$fit.",".$neckline.",".$sleeve;
        }

        $allcolors = array();
        $allsizes = array();

        $variantcounter = 0;
        $maxvariantcounter = 99; // Constraint introduced because of Shopify variant limitation
(max 100 variants per style)
        $firstvariant = true;
        foreach ($style["variants"] as $key => $variant) {
            $colorcode = $variant["ColorCode"];
            $colorname = $variant["Color"];
            $colorsequence = $variant["ColorSequence"];
            $sizecodenav = $variant["SizeCodeNavision"];
            $sizecode = $variant["SizeCode"];
            $sizesequence = $variant["SizeSequence"];
            $stock = $variant["Stock"];
            $published = $variant["Published"];
            $SKU_Start_Date = $variant["SKU_Start_Date"];
            // Take the values of the first variant to set the generic.
            if ($firstvariant) {
                $firstvariant = false;
                if ($SKU_Start_Date=="2018-01-01") {
                    $allstyles[$stylecode]->product->tags .= ",Spring/Summer 2018";
                }
                $allstyles[$stylecode]->product->published = strtoupper($published);
            }

            $price = $variant["Price<10 EUR"];
            $b2bskuref = $variant["B2BSKUREF"];
```

```
            if (!isset($allstyles[$stylecode]->product->variants)) $allstyles[$stylecode]->product->variants = array();
            $opt = new stdClass();
            $opt->option1 = $colorname;
            $opt->option2 = $sizecode;
            $opt->price = $price;
            $opt->sku = $b2bskuref;
            $opt->inventory_management = "shopify";
            $opt->inventory_policy = "continue";
            $opt->fulfillment_service = "manual";
            $opt->inventory_quantity = $stock;
            $allstyles[$stylecode]->product->variants[] = $opt;

            $allcolors[$colorname] = $colorname;
            $allsizes[$sizecode] = $sizesequence;

            $variantcounter++;
            if ($variantcounter>=$maxvariantcounter) break;
        }

        if (!isset($allstyles[$stylecode]->product->options)) $allstyles[$stylecode]->product->options = array();
        $opt = new stdClass();
        $opt->name = "Color";
        $opt->values = array();
        foreach ($allcolors as $key => $value) {
            $opt->values[] = $key;
        }
        $allstyles[$stylecode]->product->options[] = $opt;

        asort($allsizes);
        $opt = new stdClass();
        $opt->name = "Size";
        $opt->values = array();
        foreach ($allsizes as $key => $value) {
            $opt->values[] = $key;
        }
        $allstyles[$stylecode]->product->options[] = $opt;

        createProductInShopify($allstyles[$stylecode]);

        // Import images
        // 1. Retrieve product id from Shopify
        $shopifyproduct = getProductInShopifyByHandle($stylecode);
        // 2. Retrieve variant id's from Shopify
        $variantids = array();
        foreach ($shopifyproduct["variants"] as $key => $var) {
            $sku = $var["sku"];
            $colorcode = substr($sku, 7, 4);
            $id = $var["id"];
            if (!isset($variantids[$colorcode])) $variantids[$colorcode]=array();
            $variantids[$colorcode][] = $id;
        }

        // 2. Get ST/ST Pictures
        $pics = getSTSTProductImages($stylecode);

        foreach ($pics as $key => $pic) {
            $colorcode = $pic["ColorCode"];
            $picname = $pic["FName"];
            $picurl = $pic["HTMLPath"];

            $jsonData = new stdClass();
            $jsonData->image = new stdClass();
            $jsonData->image->src = $picurl;
            // Get the first char of the filename : if SFM -> position 1
            if (substr($picname,0,3)=="SFM") {
                $jsonData->image->position = "1";
            }
            $jsonData->image->variant_ids = $variantids[$colorcode];

            createProductImageInShopify($shopifyproduct["id"], $jsonData);
        }

    } else {
        // Similar logic for the update
        // ...
```

```
    }
}
```