Amasty

For more details see the Multi Warehouse Inventory extension page.

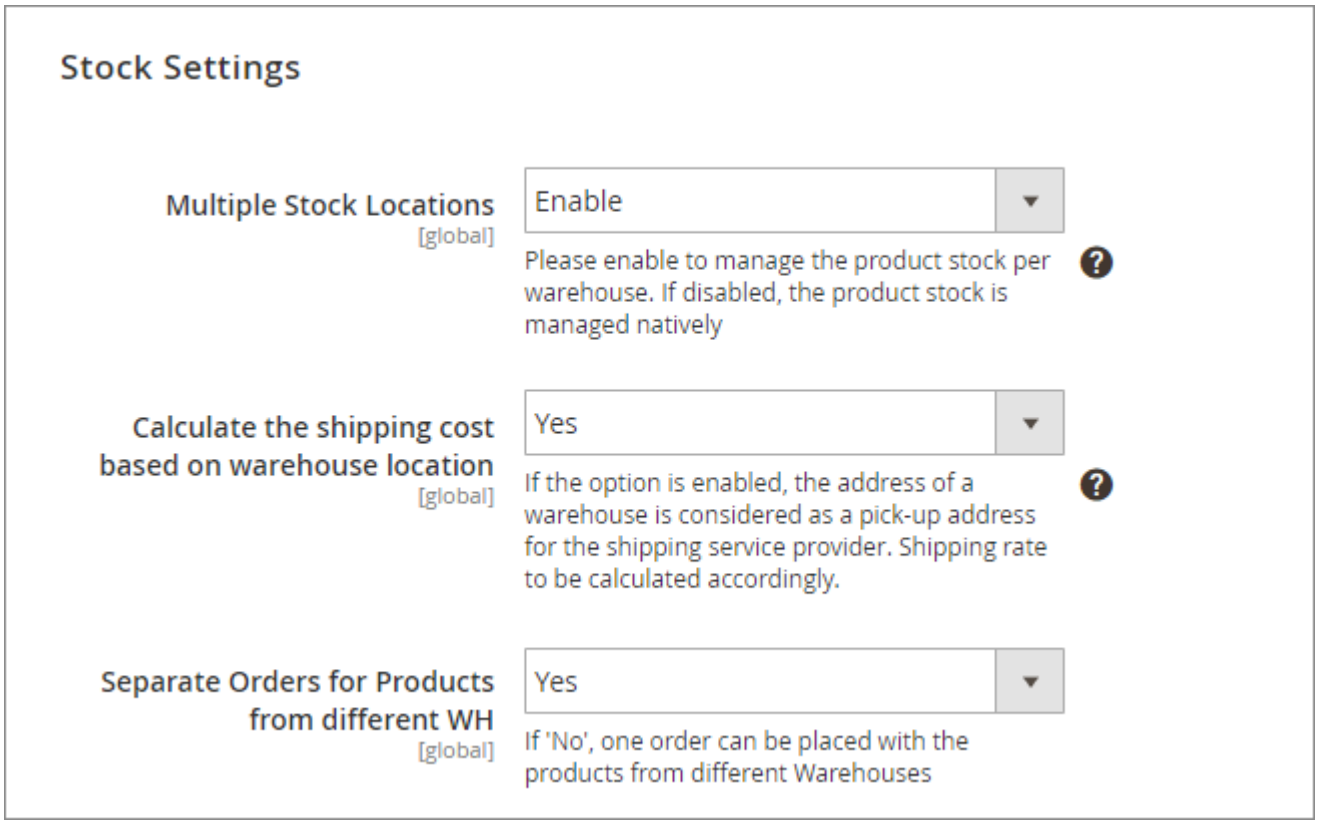
Multi Warehouse Inventory

Split inventory among multiple warehouses and keep track of stock items on a website or a store view level. Manage stock data for each separate warehouse on one handy grid.

- Distribute inventory among different warehouses
- Assign warehouses to store views and customer groups
- Use a smart algorithm for warehouse selection
- Set a default warehouse for unassigned items
- Separate orders from different warehouses

Stock Settings

To configure general and stock settings please go to **Stores → Configuration → Multi Warehouse Inventory**.

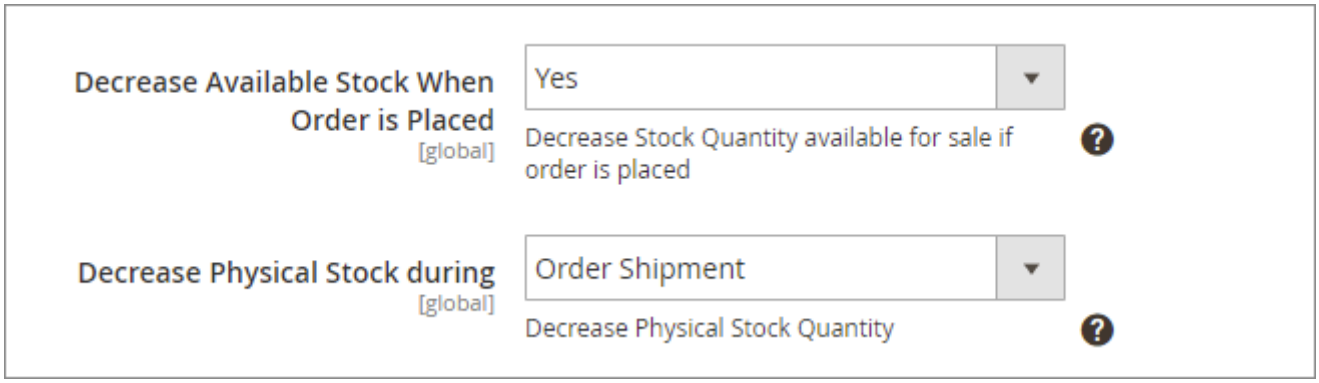


Multiple Stock Locations - Set the option to *Yes* to enable the extension.

Calculate the Shipping cost based on warehouse location - Set the option to *Yes* to make the warehouse address a primary pickup address for the shipping service provider. With the option disabled, the store address, that is mentioned in the config, remains a primary pickup address.

When the ordered products are assigned to multiple warehouses: either multiple orders may be created, or one order with the shipping rates calculated separately for each warehouse and summarized.

Separate Orders from Different Warehouses - in case one order contains products from different warehouses, the module will divide them into several shipments and calculate delivery costs for each shipment separately depending on the warehouse location.



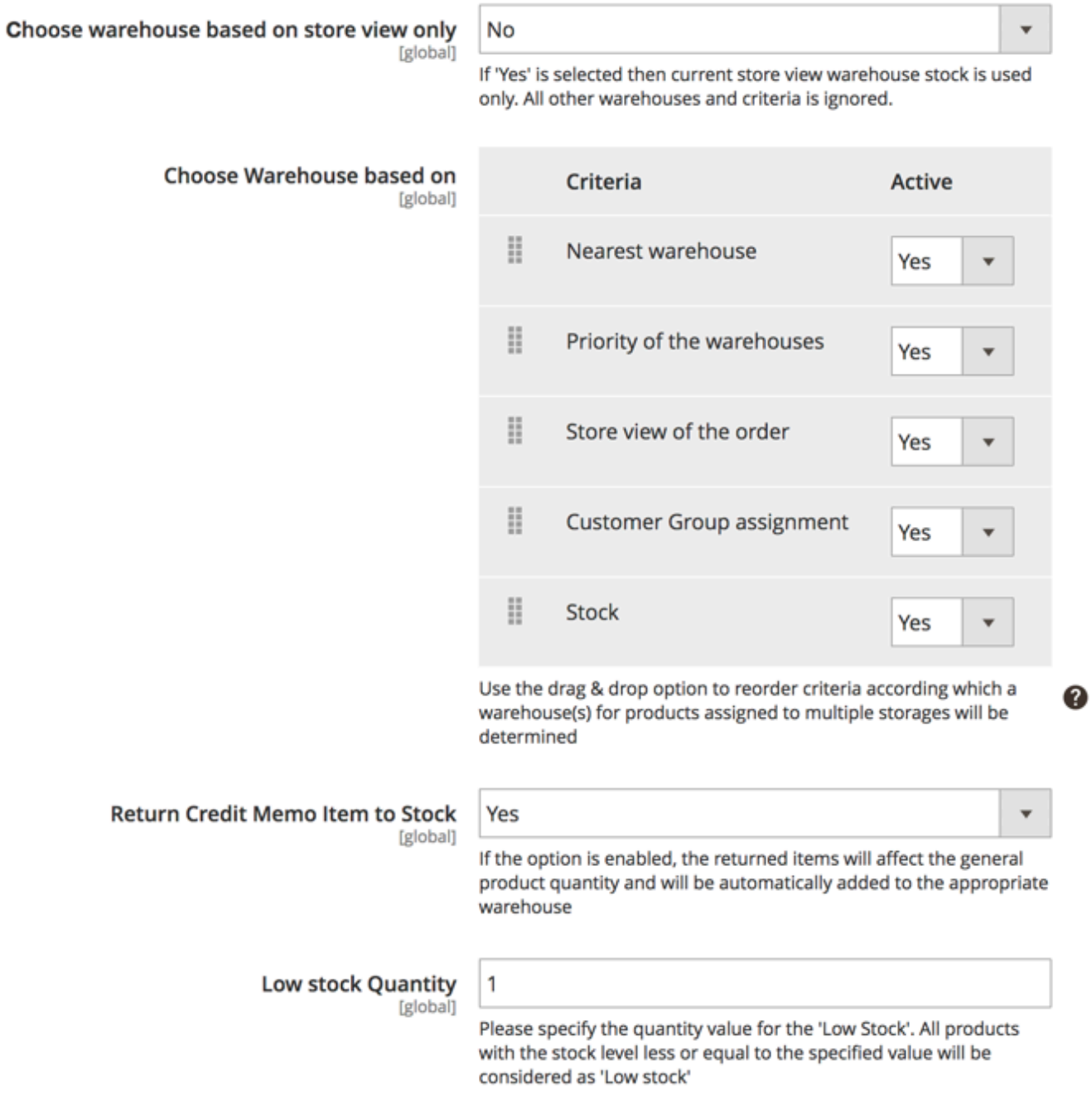
Decrease Available Stock When Order is placed - If the option is set to 'yes', the available stock will be automatically decreased once an order is placed (Available Qty = Physical Stock - Qty to ship). If the option is disabled 'Available Quantity' will be equal to 'Physical Quantity'(Physical Qty = Available Qty + Qty to Ship).

Available Qty = Physical Qty (are identical) when 'Decrease Available Qty When Order is placed'is set to 'No'.

Decrease physical stock during- Choose the condition to decrease Physical Quantity (Physical Qty = Available Qty + Qty to Ship).

- Order Shipment (default);
- Order Creation (available only when 'Decrease Available Qty When Order is Placed' is set to 'Yes');
- Invoice creation;

If you choose to **decrease Physical Qty** during **Order Creation**, the **Available Qty** will be equal to **Physical Qty**



Choose warehouse based on store view only - Enable this option to use the stock only of the current store view warehouse. All other warehouses and criteria will be ignored.

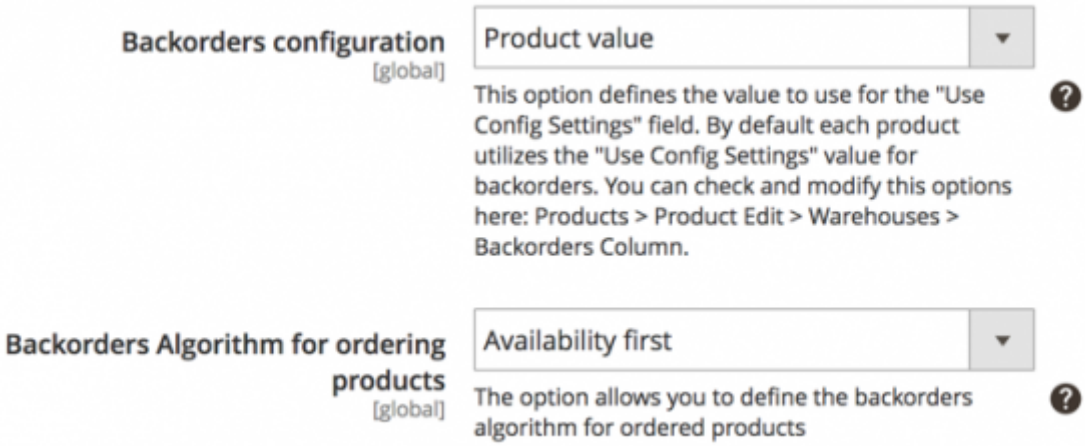
Place the Stock criteria on top and enable it to select warehouse based on available product quantity and avoid situations when a customer has to receive the order in parts from multiple warehouses.

Choose Warehouse based on - define the warehouse selection algorithm that will be used when one product is assigned to multiple warehouses. Drag and drop criteria to place them in the order you need.

NOTE: If all the criteria are set to *No*, the extension selects the warehouse, that has the biggest inventory.
NOTE: To get more details on how the algorithm works please see the 'Order Shipment' section.

Return Credit Memo Item to Stock - if the option is enabled, returned items will be automatically added to stock of the appropriate warehouse.

Low Stock Quantity - specify stock quantity that will be considered as a low stock.



Backorders configuration — choose Product value or Warehouse value options:

- Product Value — the backorders value will be used from the product's Advanced Inventory;

Please note that by default backorders are not available. Go to product page and unflag the Use Config Settings box to allow backorders.

- Warehouse value — the global backorders value will be used for each particular warehouse.

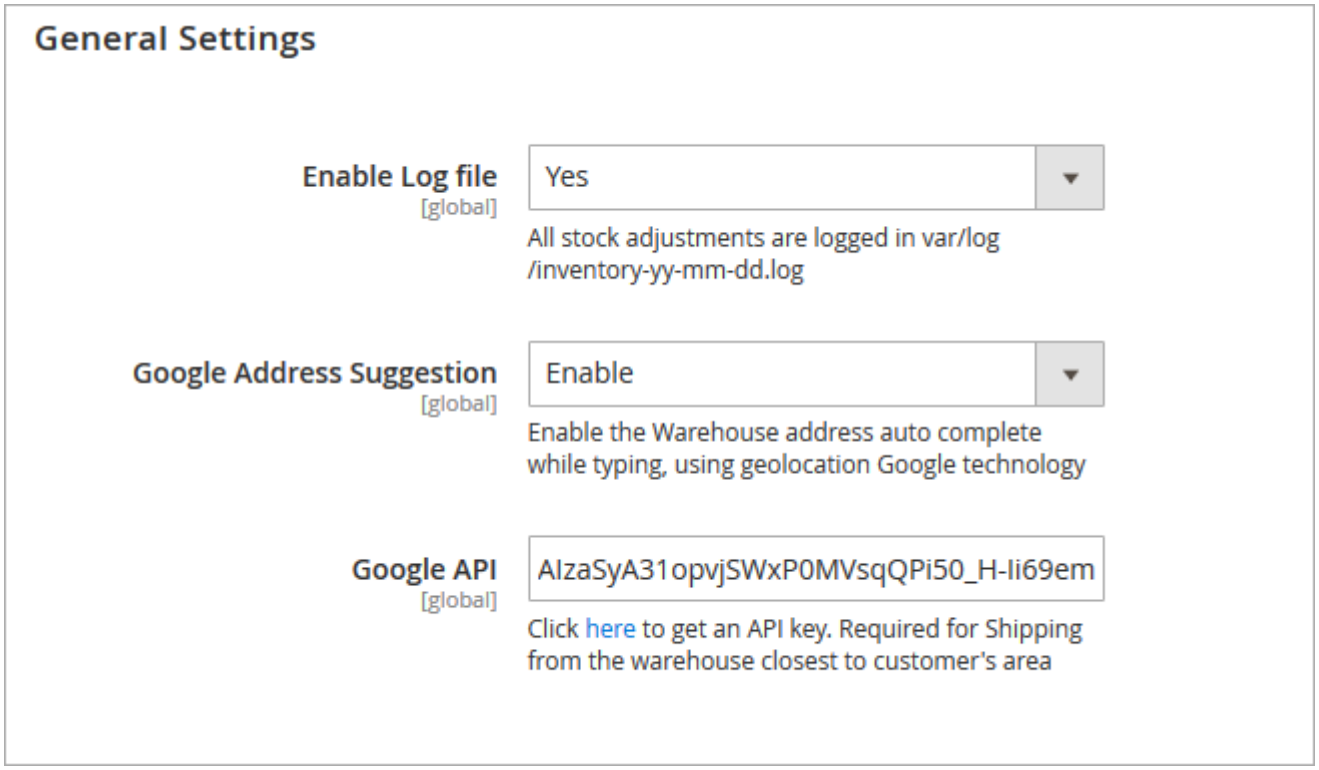
This value can be set from the Products in Warehouse section (Products → Warehouses → Warehouse). There you can change the backorders settings for each particular product in the warehouse.

Backorders Algorithm for ordering products — define the primarly rule for using backorders:

- Availability first — choose this option to use backorders instead of selling the products that are available in other warehouses;
- Backorders first — choose this option to use backorders only when the product is not available in all warehouses.

Please see the Backorders section of this guide to configure this functionality.

General Settings



Enable Log File - Log all stock adjustments.

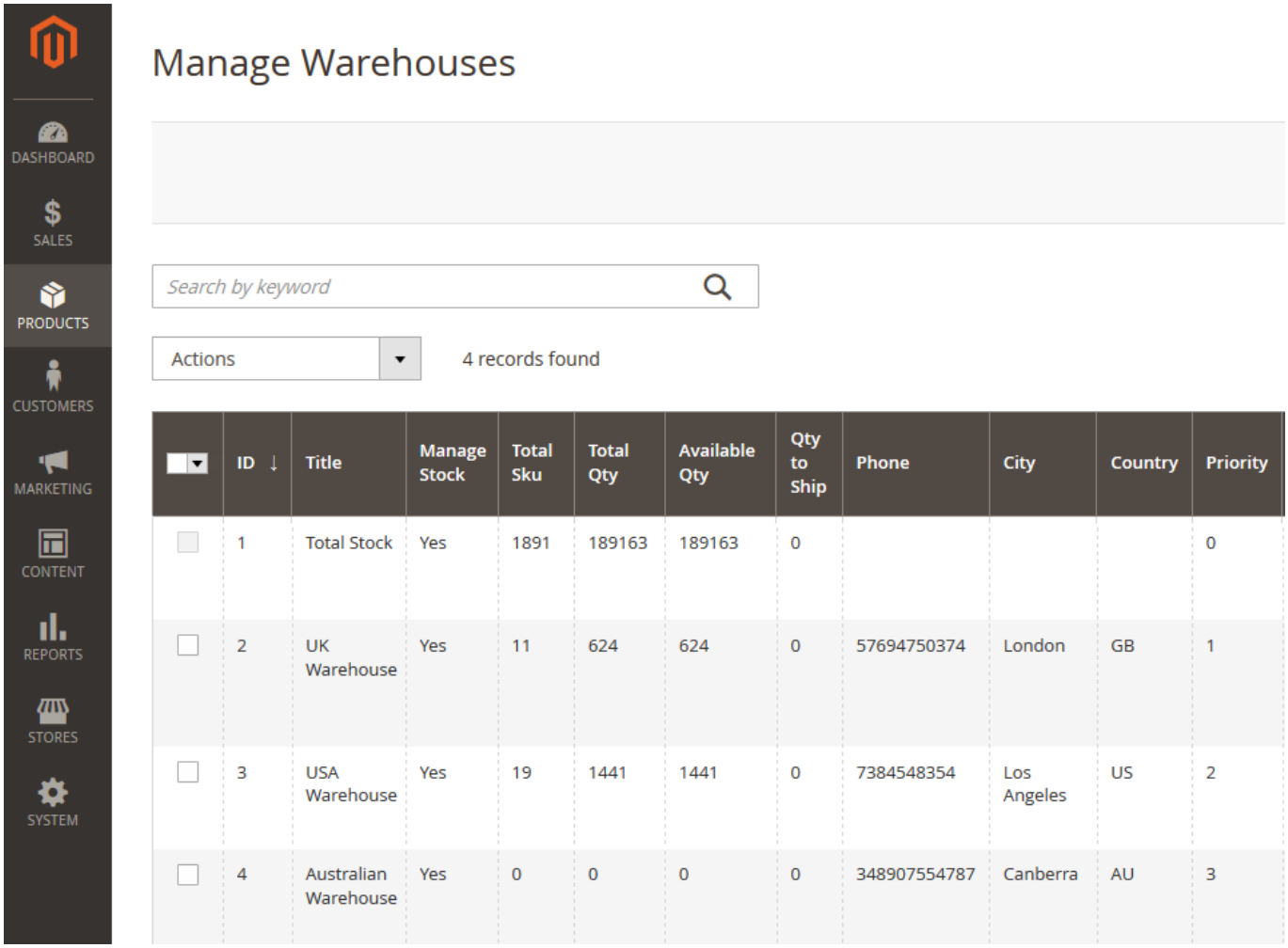
Google Address Suggestion - Enable this option to automatically fill in address fields while creating a warehouse.

Google API - This option will automatically determine the warehouse closest to customer's location.

Please click here to get the **Google API Key**.

Manage Warehouses

To create or edit already created warehouses go to **Products → Inventory → Warehouses**.



ID	Title	Manage Stock	Total Sku	Total Qty	Available Qty	Qty to Ship	Phone	City	Country	Priority
1	Total Stock	Yes	1891	189163	189163	0				0
2	UK Warehouse	Yes	11	624	624	0	57694750374	London	GB	1
3	USA Warehouse	Yes	19	1441	1441	0	7384548354	Los Angeles	US	2
4	Australian Warehouse	Yes	0	0	0	0	348907554787	Canberra	AU	3

Warehouses Grid - All created warehouses are displayed on a special grid where admin users can see the necessary warehouse information including stock and address data, store views and customer groups.

Total Quantity - shows the total number of products stored in a particular warehouse at the current moment (Available quantity + Quantity to Ship).

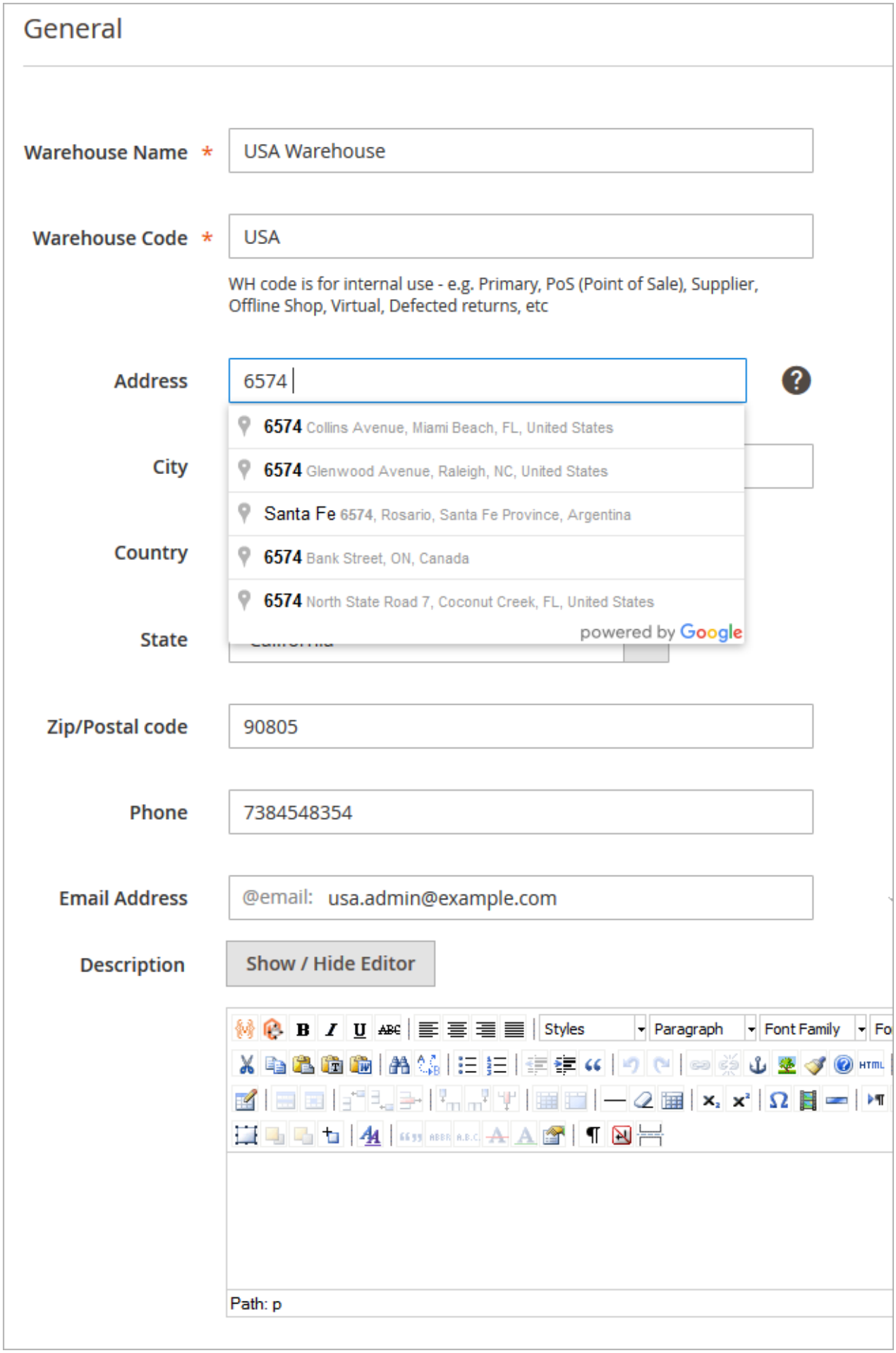
Available Quantity - shows the number of products in a warehouse available for orders (Total Quantity-Quantity to Ship).

Quantity to Ship - show the number of ordered items ready to be shipped.

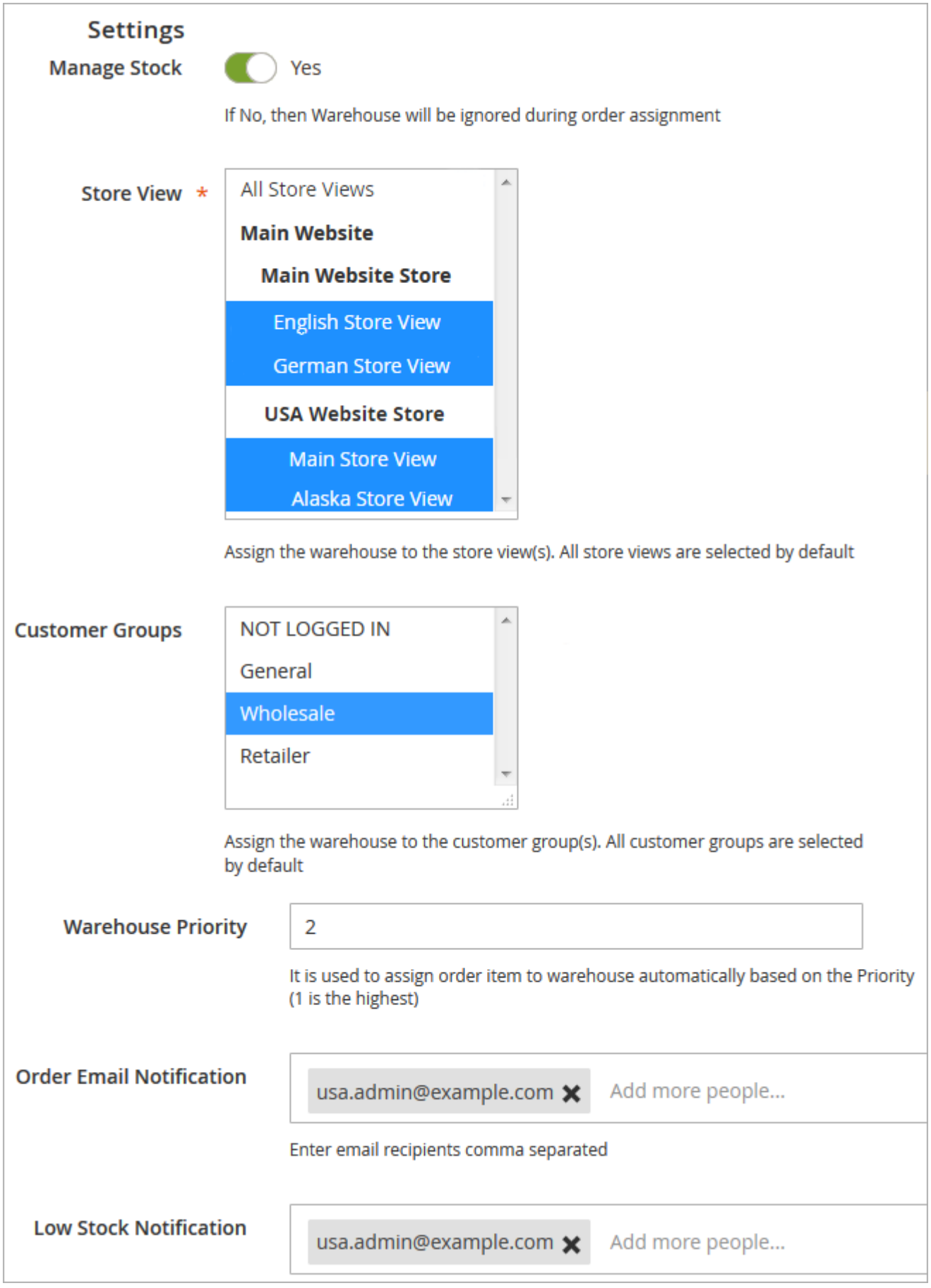
The module lets you **duplicate**, **edit** and **delete** already created warehouses. To create a new warehouse please click on the **'Add Warehouse'** button.

Warehouse Creation: Warehouse Info

Specify warehouse information including address data and additional description. Use **Google Address Auto Suggestion** option when filling in address fields.


General
Warehouse Name *
USA Warehouse
Warehouse Code *
USA
WH code is for internal use - e.g. Primary, PoS (Point of Sale), Supplier, Offline Shop, Virtual, Defected returns, etc
Address
6574
6574 Collins Avenue, Miami Beach, FL, United States
6574 Glenwood Avenue, Raleigh, NC, United States
Santa Fe 6574, Rosario, Santa Fe Province, Argentina
6574 Bank Street, ON, Canada
6574 North State Road 7, Coconut Creek, FL, United States
powered by Google
City
Country
State
Zip/Postal code
90805
Phone
7384548354
Email Address
@email: usa.admin@example.com
Description
Show / Hide Editor
Styles
Paragraph
Font Family
Path: p

Warehouse Creation: Settings



Store View - Assign the warehouse to a specific store view if necessary.

One and the same warehouse can be assigned to multiple stores and views.

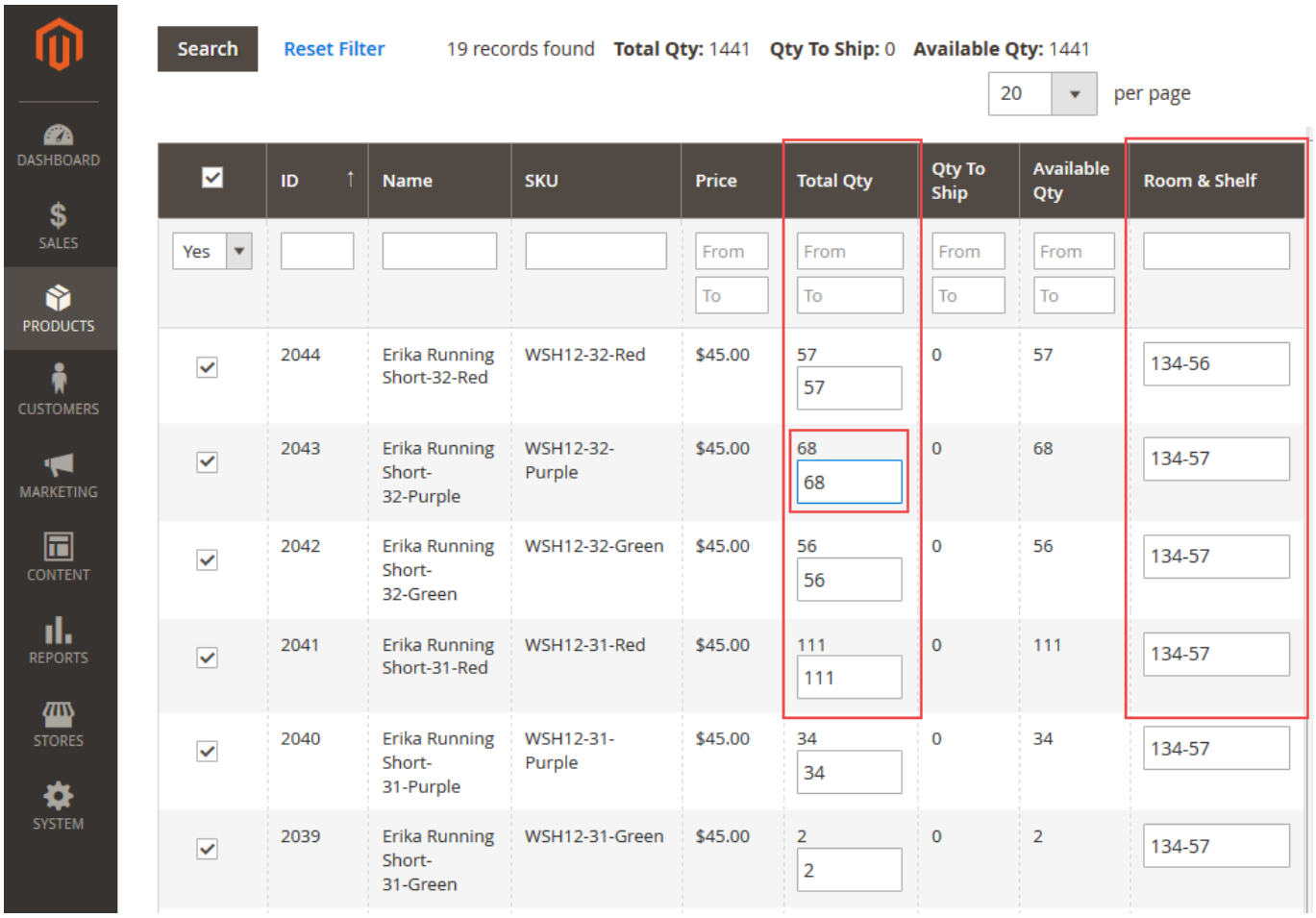
Customer Groups - Associate the warehouse with a specific customer group(s)

Warehouse Priority - Set the warehouse priority. The option will be used to assign a product to a warehouse depending on its priority (1 is the highest priority).

Order Email Notifications and **Low Stock Email Notifications** - specify email addresses to which notifications will be sent.

Assign Products to a Warehouse

You can either assign all necessary products to a warehouse on the corresponding warehouse page or add a specific product to multiple warehouses from a product page.

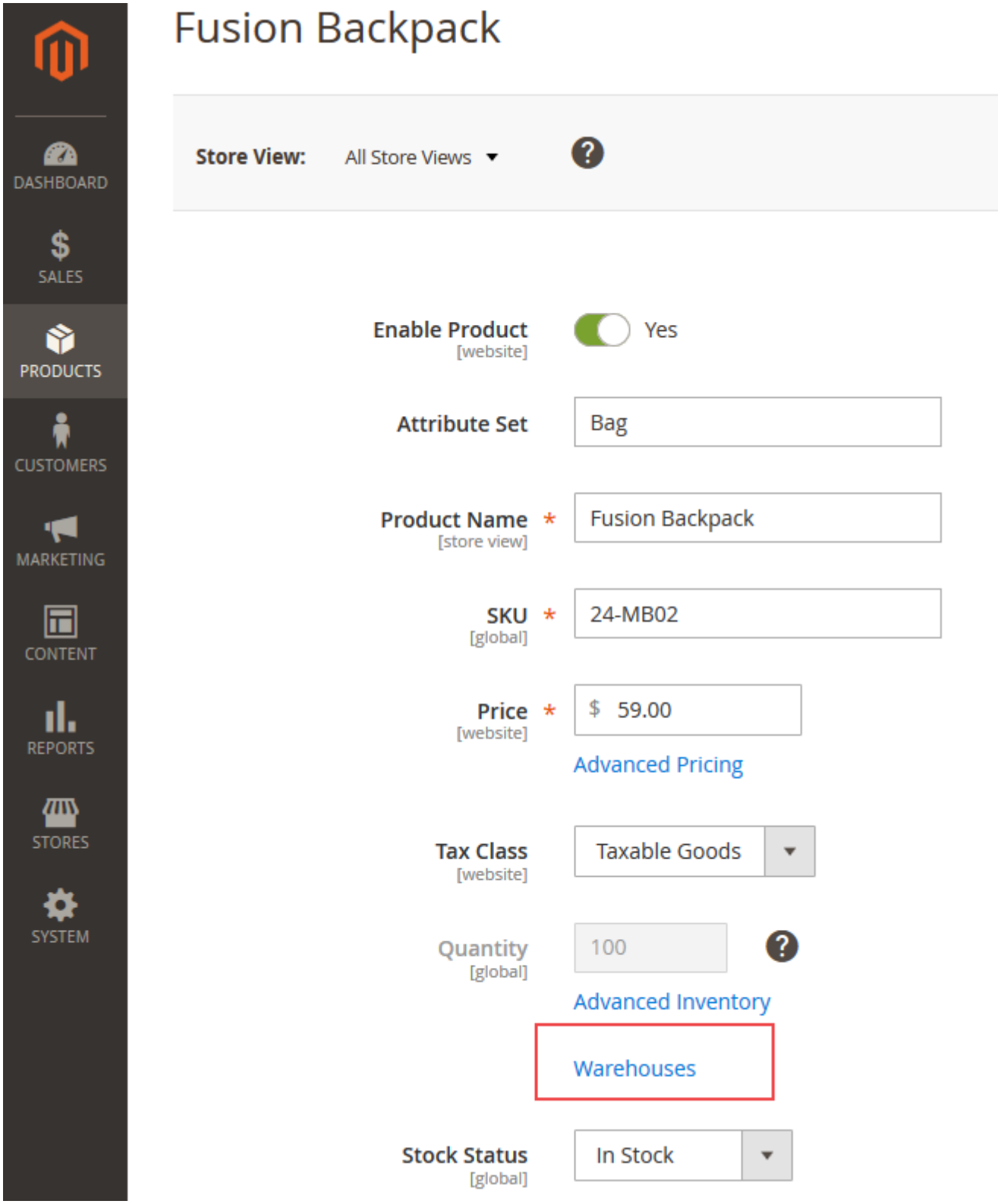


All **out of stock products** are automatically removed from the product listing.

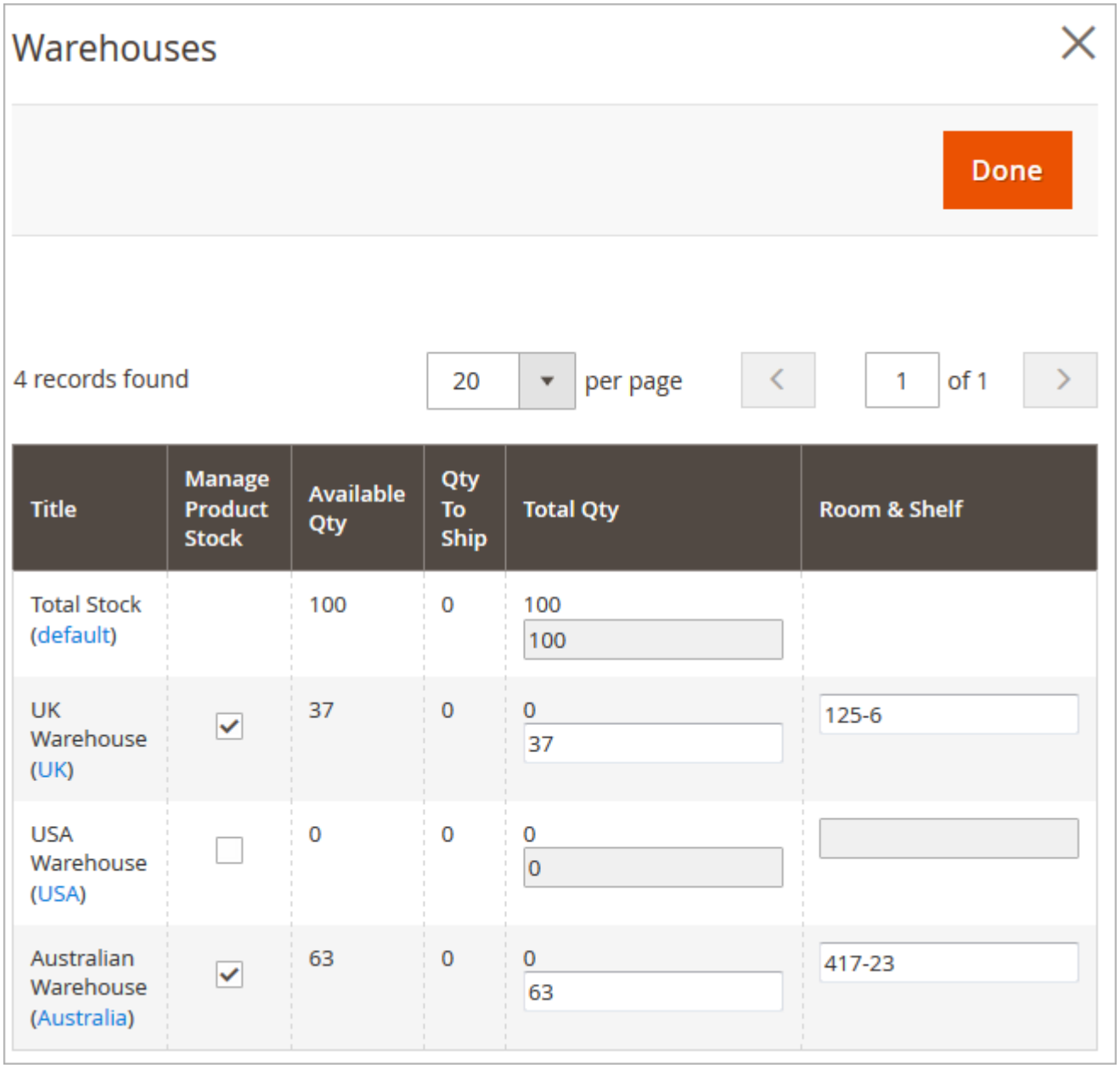
Assign products on the warehouse settings page - tick items that should be added to the warehouse and edit their stock quantity (total quantity and quantity to ship) right on the grid. Specify a room and a shelf info for each item if necessary.

To assign items from a product page to multiple warehouses, please go to **Products → Catalog →**

open the product you need and click on the 'Warehouses' link.



On this page you can assign the product to necessary warehouses, edit their stock and specify a room & a shelf.



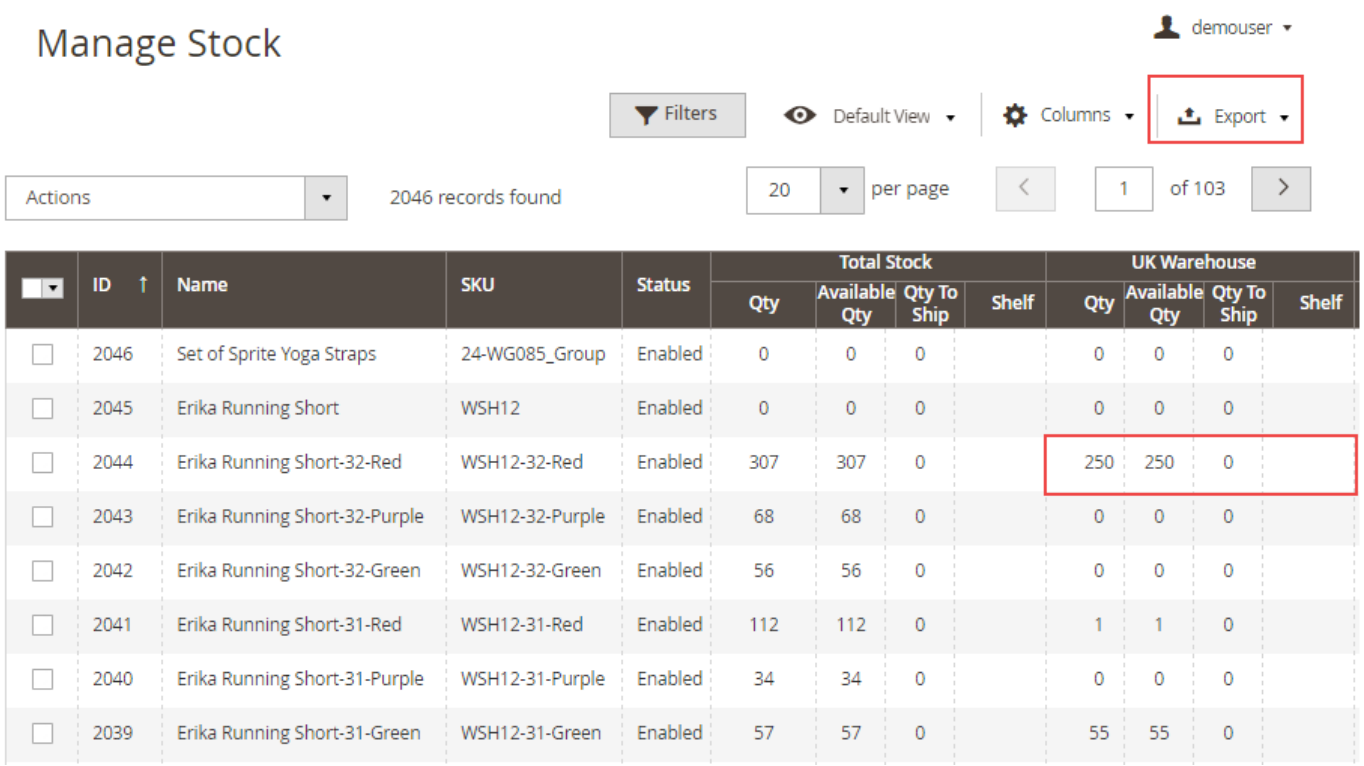
Shipping Methods and Rates

Use custom shipping rates - To set your custom shipping methods and rates (different from the ones set by shipping providers), enable the option.

Shippings - Set your custom rates for each shipping method.

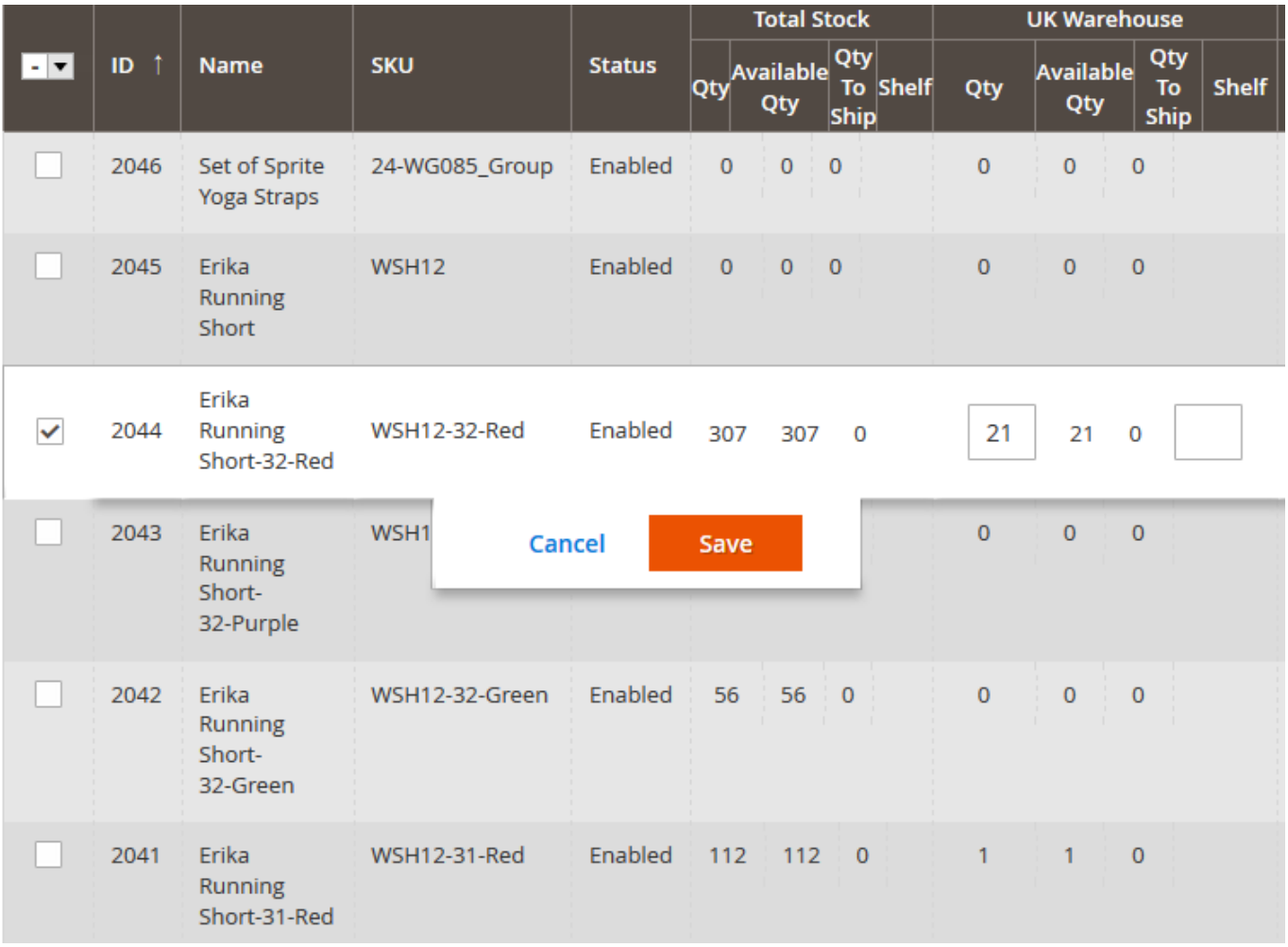
- *To set free shipping - enter zero.*
- *Leave empty to use the shipping providers rates.*

Manage Stock

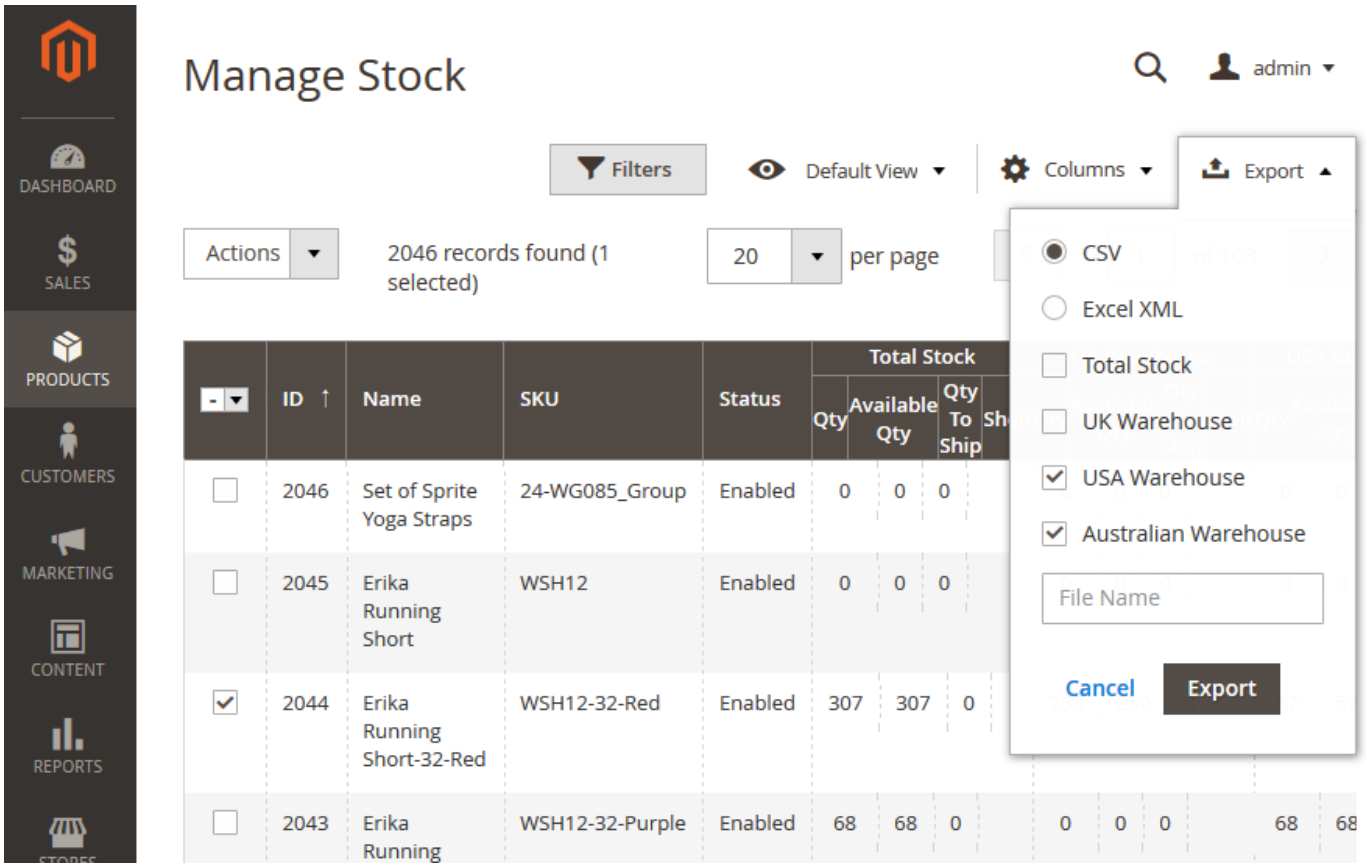


	ID	Name	SKU	Status	Total Stock				UK Warehouse			
					Qty	Available Qty	Qty To Ship	Shelf	Qty	Available Qty	Qty To Ship	Shelf
☐	2046	Set of Sprite Yoga Straps	24-WG085_Group	Enabled	0	0	0		0	0	0	
☐	2045	Erika Running Short	WSH12	Enabled	0	0	0		0	0	0	
☐	2044	Erika Running Short-32-Red	WSH12-32-Red	Enabled	307	307	0		250	250	0	
☐	2043	Erika Running Short-32-Purple	WSH12-32-Purple	Enabled	68	68	0		0	0	0	
☐	2042	Erika Running Short-32-Green	WSH12-32-Green	Enabled	56	56	0		0	0	0	
☐	2041	Erika Running Short-31-Red	WSH12-31-Red	Enabled	112	112	0		1	1	0	
☐	2040	Erika Running Short-31-Purple	WSH12-31-Purple	Enabled	34	34	0		0	0	0	
☐	2039	Erika Running Short-31-Green	WSH12-31-Green	Enabled	57	57	0		55	55	0	

To edit stock data please go to **Products → Inventory → Manage Stock**.



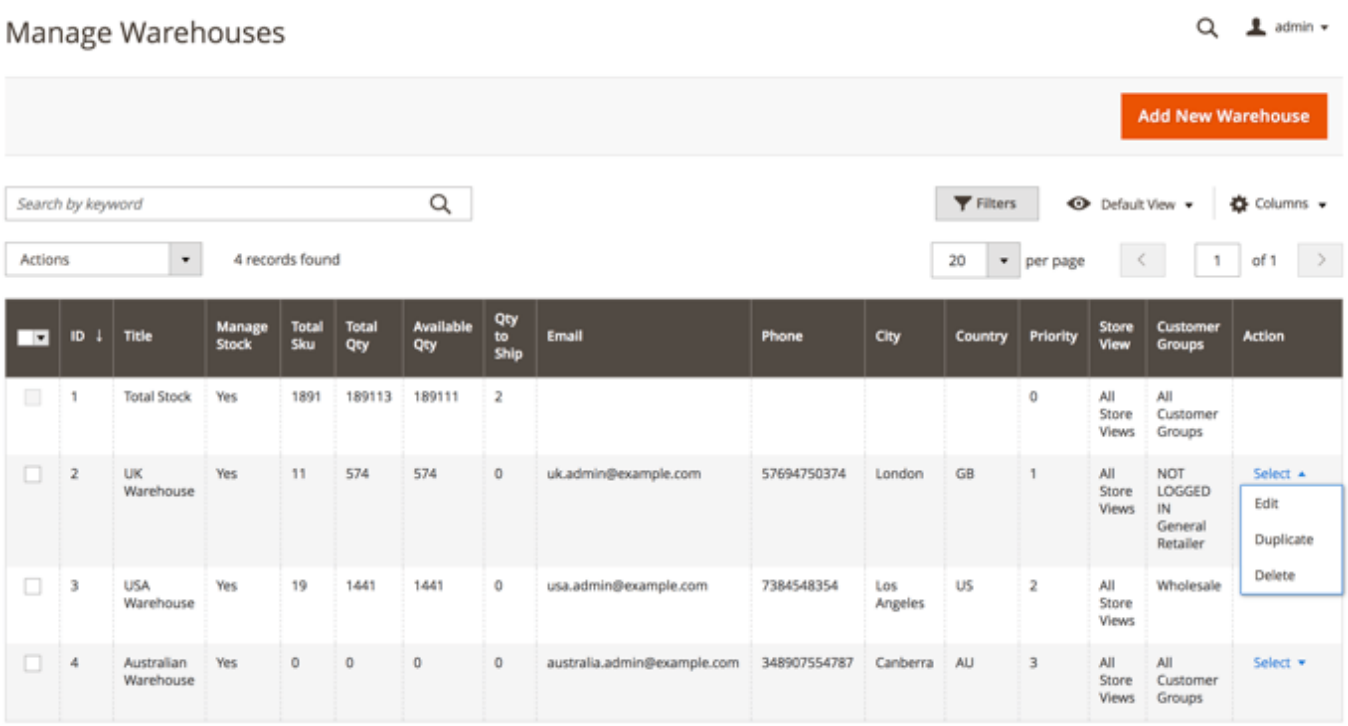
Manage Stock Grid - To edit stock quantity please select the necessary warehouse on the grid and click on the box with the value you want to change.



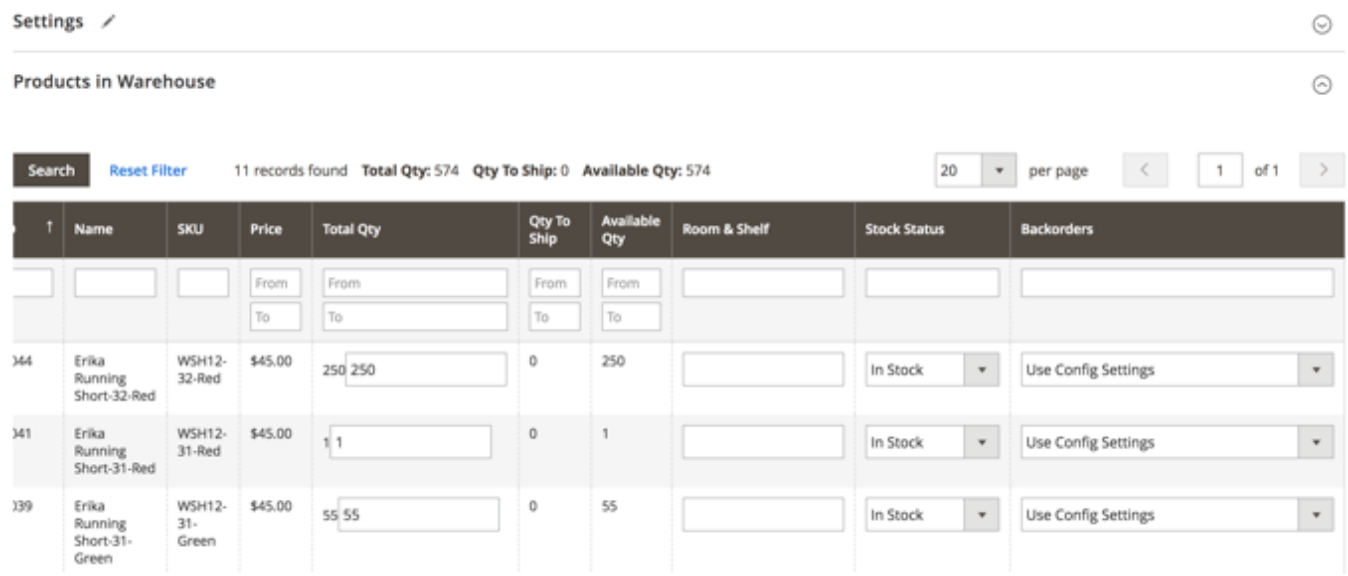
Export Stock data - You can export stock data right from the grid. It's possible to export data from all warehouses at once or choose specific warehouses.

Backorders

Backorders functionality allows you to sell products that are currently out-of-stock in your warehouses and provides customers with the ability to continue placing orders when the product is unavailable. With the extension, you can configure various backorders scenarios according to your needs. This section will guide you through the process of backorders configuration.

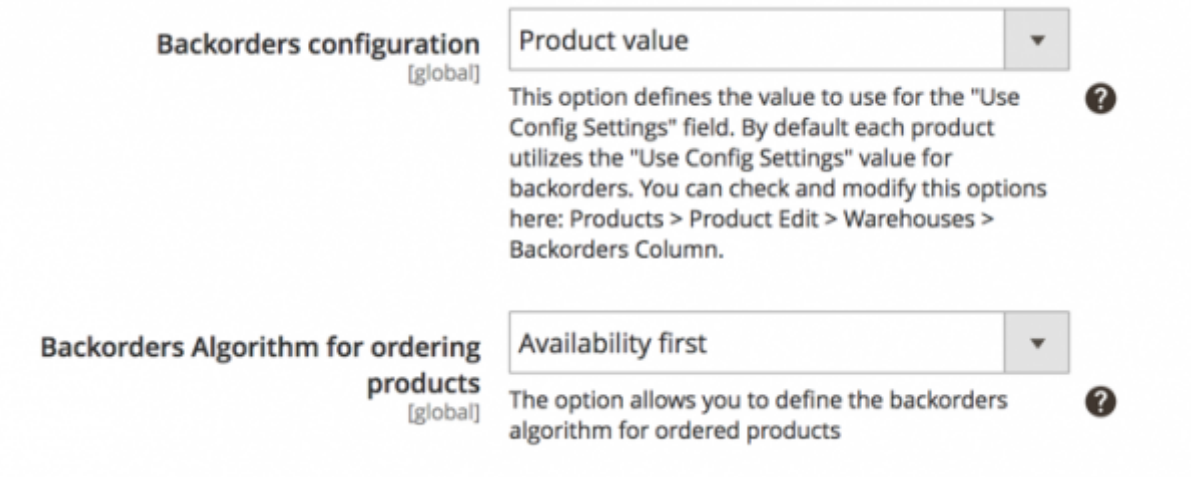


First, you need to configure the global backorders policy for each particular warehouse. Navigate to **Products → Warehouses** and choose one of the available warehouses to edit.



Expand the **Products in Warehouse** tab. By default options in the backorders column are switched to **Use Config Settings**. It means that the global backorder behavior is set using the **Backorders configuration** option.

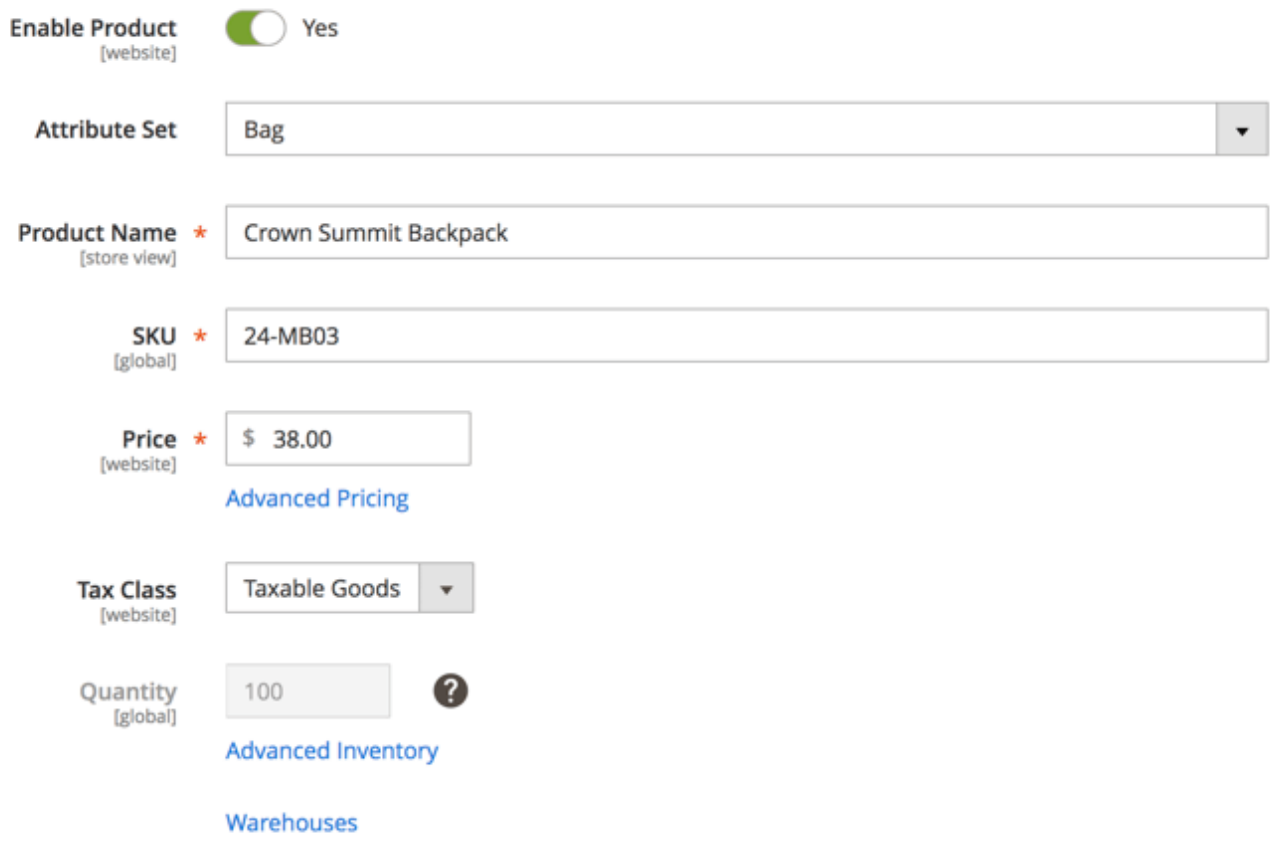
To enable the backorders functionality for selected products in certain warehouses, please modify the Backorders column for a certain product (disallow backorders or allow qty below 0).



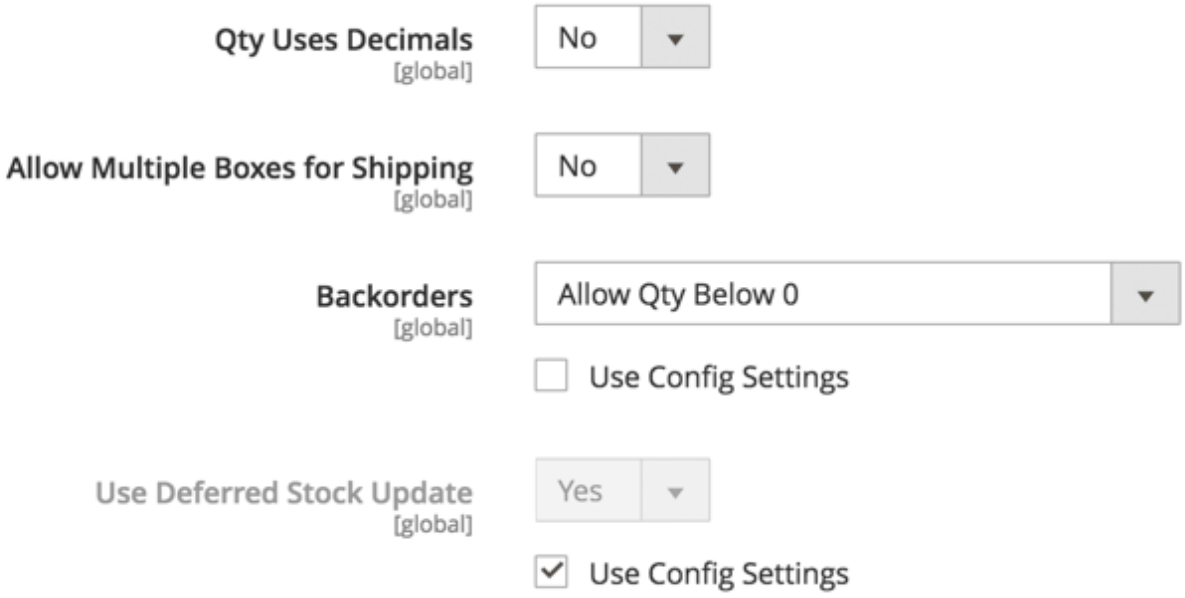
To configure it, go to **Stores → Configuration → Amasty Extensions → Multi Warehouse Inventory → Stock Settings** and modify the **Backorders configuration** option:

- choose **Product value** to configure backorders for each particular product and apply this rule to all warehouses;
- choose **Warehouse value** to configure backorders for all products in a single warehouse separately from other warehouses.

Product value



If you selected the **Product value** option, go to **Products → Catalog** and select the product you want to add the backorders functionality to. Find the **Advanced Inventory** link and click it. Then, unflag the **Use Config Settings** box under the **Backorders** option.

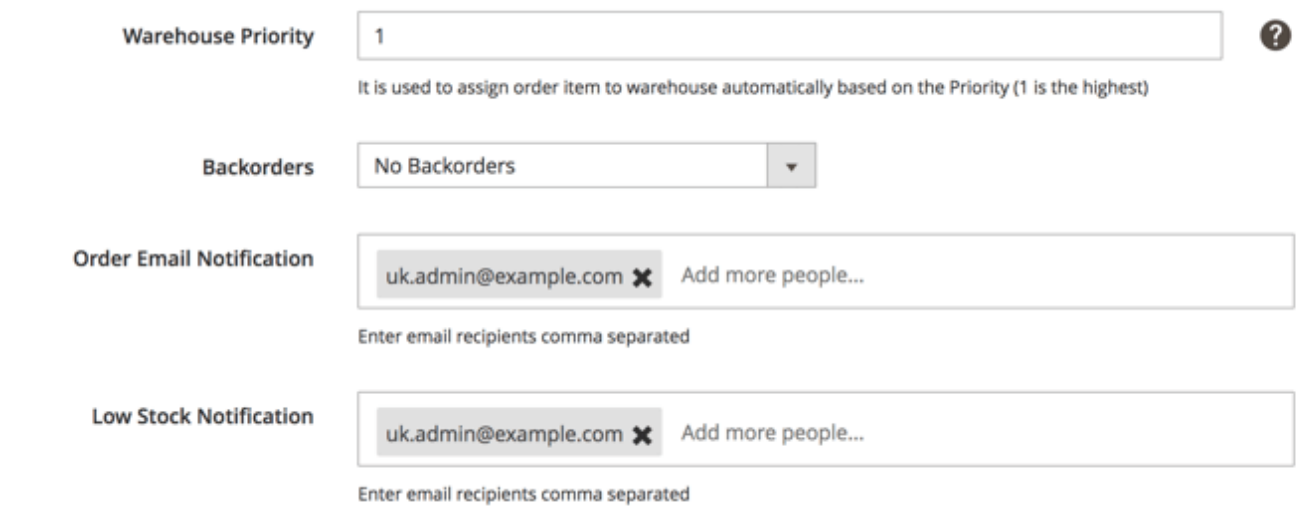


Now, you can modify the option and choose **Allow Qty Below 0** or **Allow Qty Below 0 and Notify Customer**.

This way you can add backorders for required products and this functionality will be added for all warehouses.

Warehouse value

If you selected the **Warehouse value** option, go to **Products → Warehouses** and select one of the warehouses you want to add the backorders functionality to. Expand the **Settings** section and modify the **Backorders** option by choosing **Allow Qty Below 0** or **Allow Qty Below 0 and Notify Customer**.

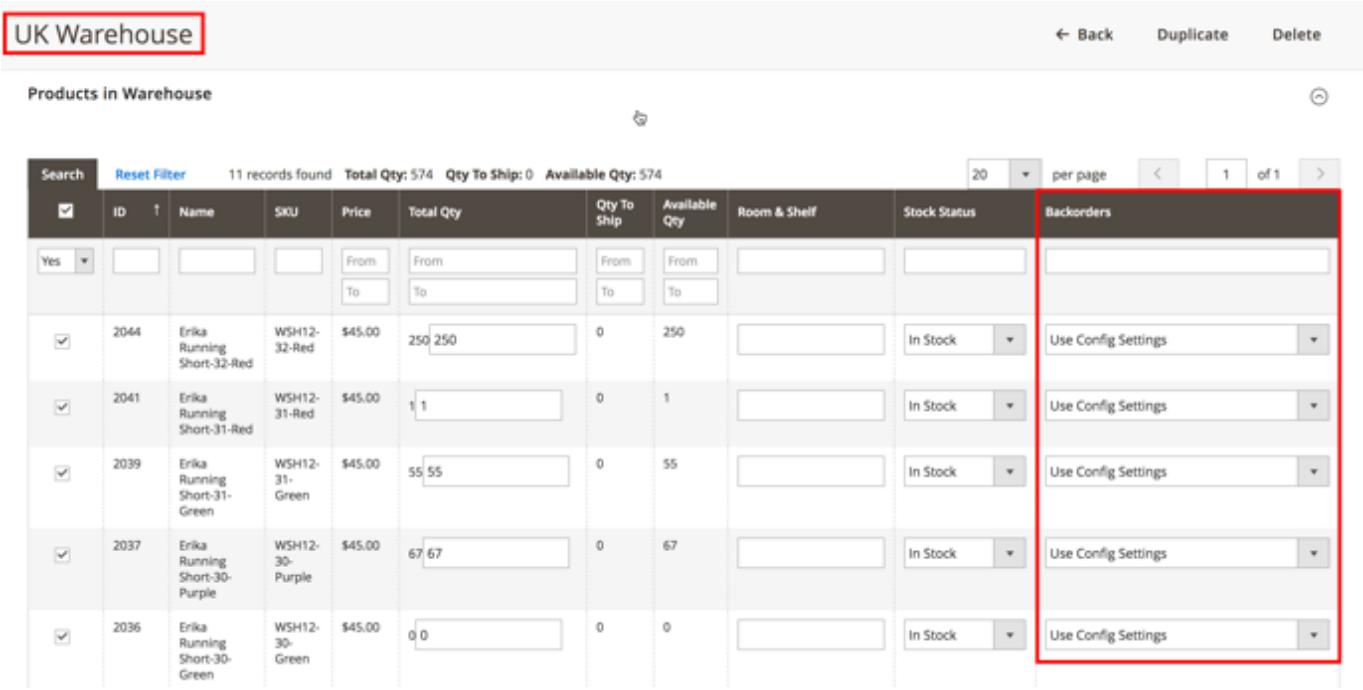


Now, the backorders functionality is available for all products assigned to this warehouse. Modify the same settings for other warehouses if needed.

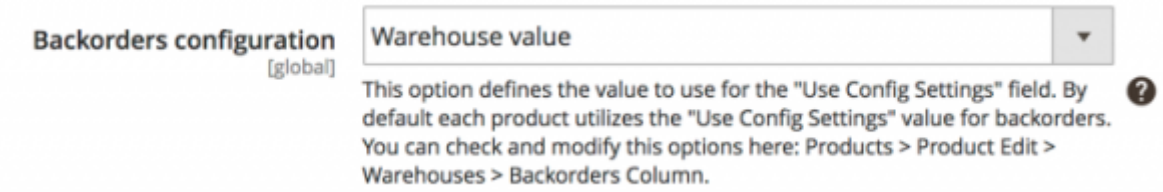
Backorders configuration examples

1. Enable backorders for all products in particular warehouses

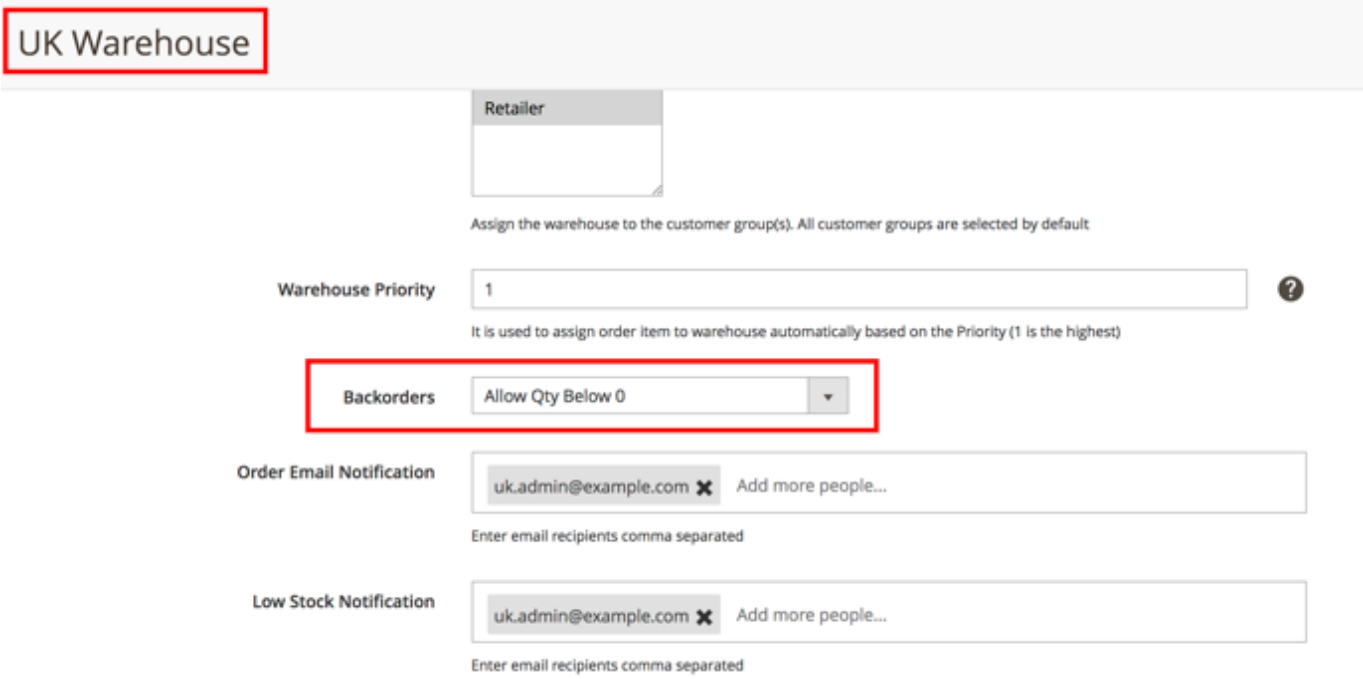
- Go to **Products → Warehouses**, select the required warehouse, and expand the **Products in Warehouse** section. Please make sure that the all products in the Backorder column have the **Use Config Settings** option selected.



- Go to **Stores → Configuration → Amasty Extensions → Multi Warehouse Inventory → Stock Settings** and switch the Backorders configuration option to **Warehouse value**.



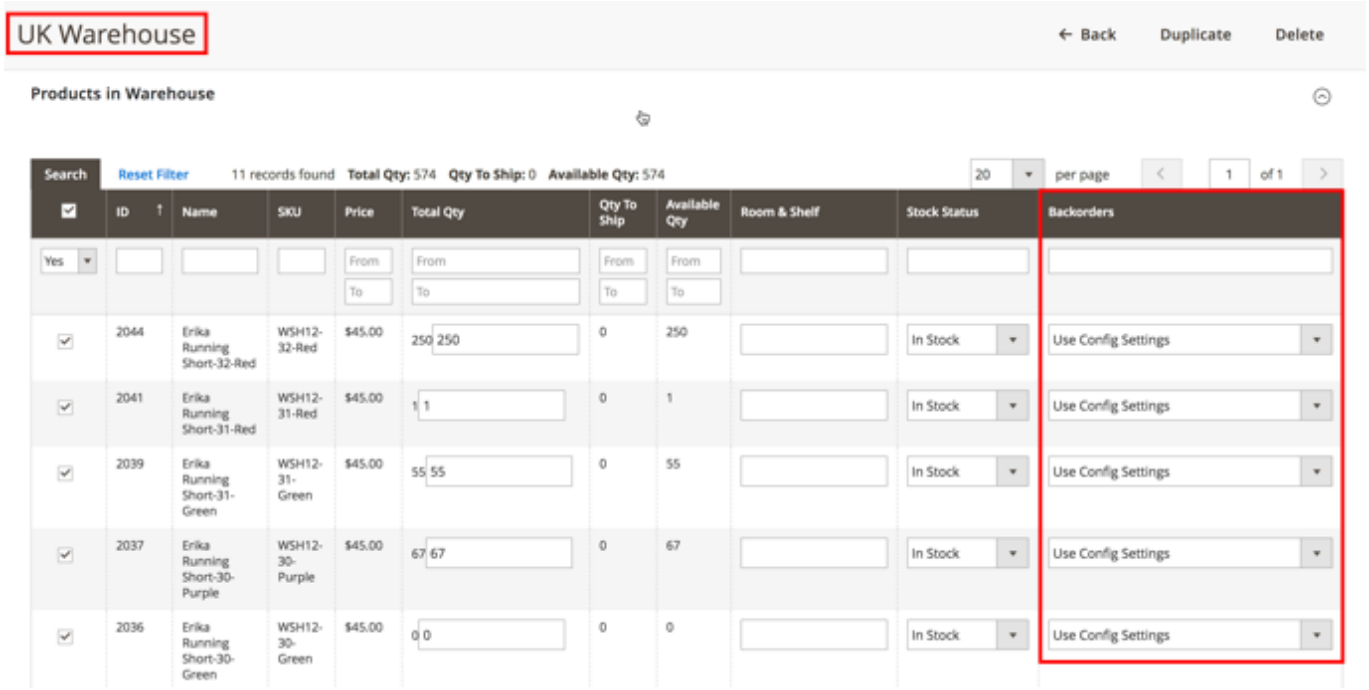
- Return to the required warehouse in the **Products → Warehouses** menu and expand the Settings section. Switch the Backorders option to **Allow Qty Below 0** or **Allow Qty Below 0 and Notify Customer**.



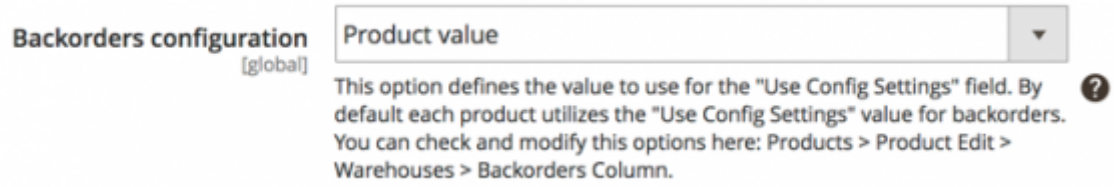
- You've enabled backorders for all products in this warehouse.

2. Enable backorders for particular products in all warehouses

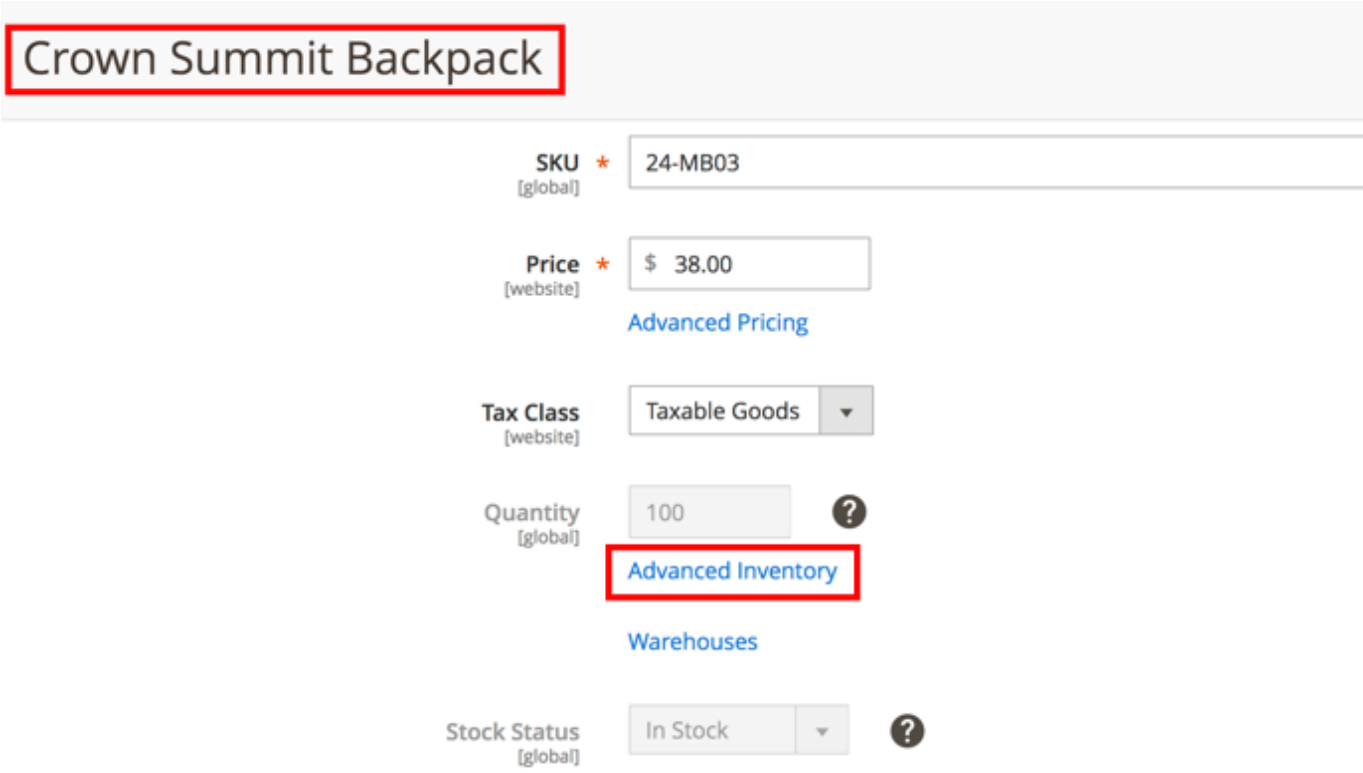
- Go to **Products → Warehouses**, select the required warehouse, and expand the **Products in Warehouse** section. Please make sure that the all products in the Backorder column have the **Use Config Settings** option selected.



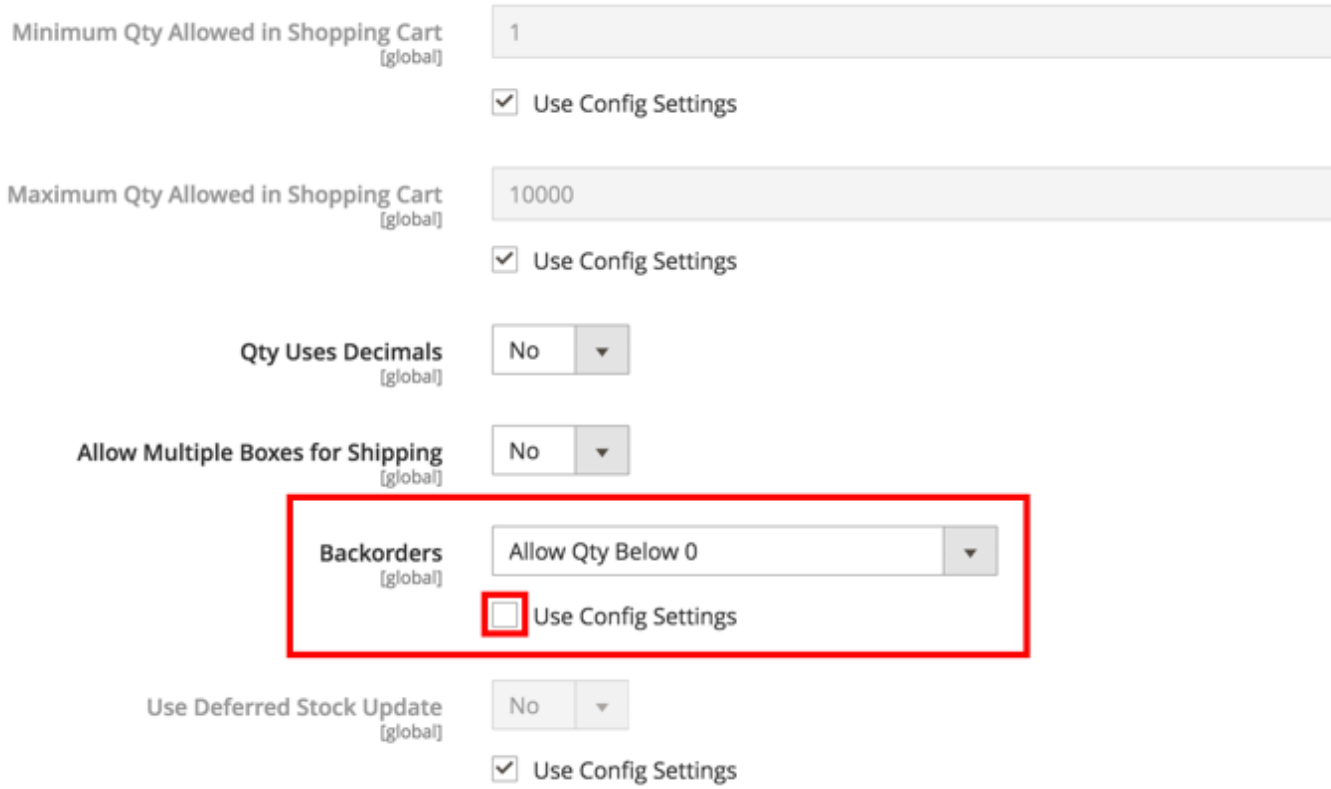
- Go to **Stores → Configuration → Amasty Extensions → Multi Warehouse Inventory → Stock Settings** and switch the Backorders configuration option to **Product value**.



- Go to Products → Catalog and select the required product. Click the **Advanced Inventory** link and then unflag the **Use Config Settings** box under the **Backorders** option.



- Switch the **Backorders** option to **Allow Qty Below 0** or **Allow Qty Below 0 and Notify Customer**. Repeat this operation for all products you want to enable backorders for.

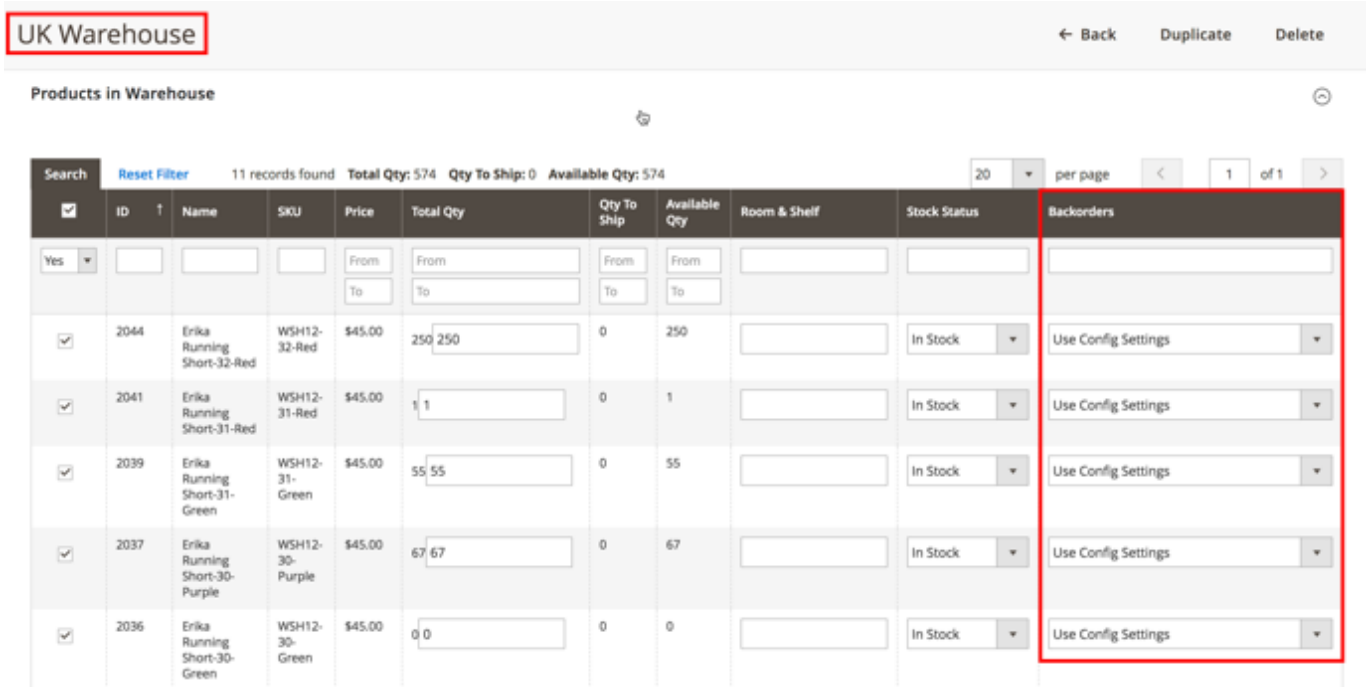


- You've enabled backorders for required products in all warehouses.

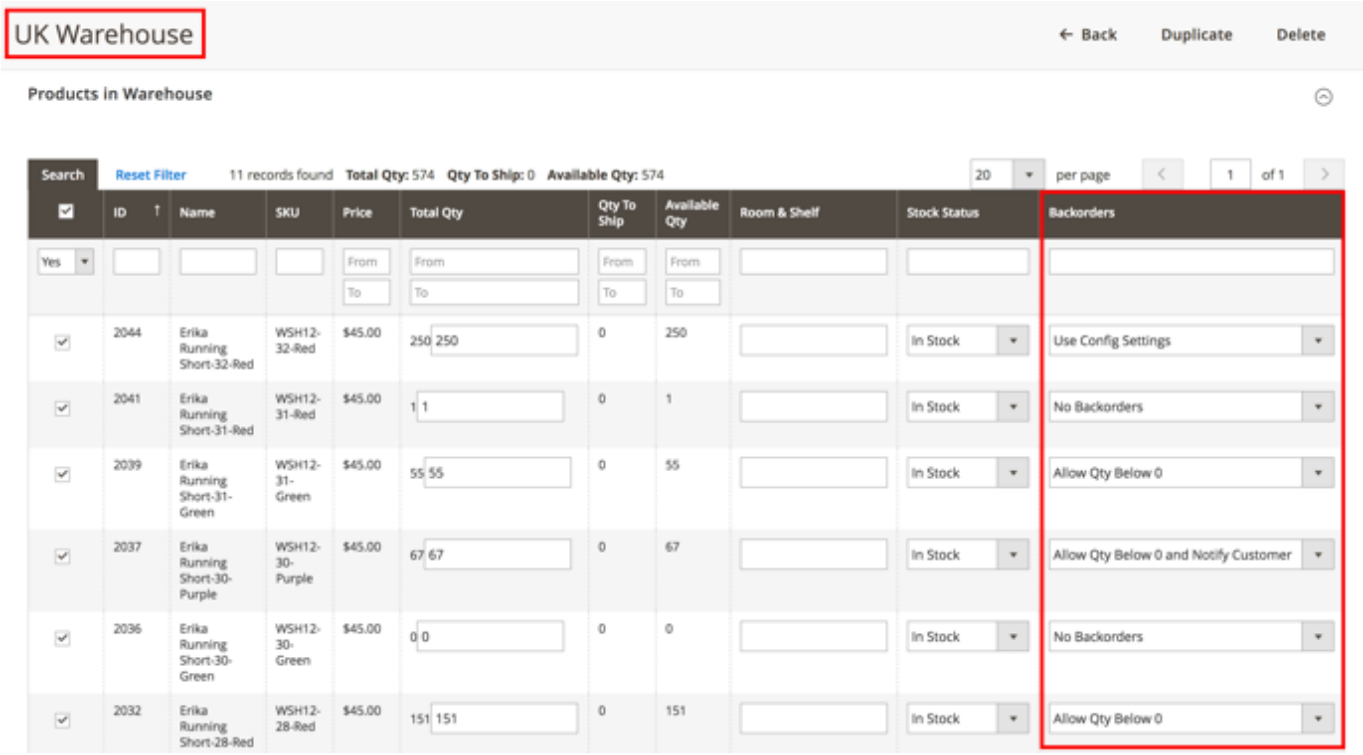
3. Set custom backorders behavior for certain products in selected warehouses

This example will help you to provide selected products in certain warehouses with individual backorders settings.

- Go to **Products → Warehouses**, select the required warehouse, and expand the **Products in Warehouse** section. For the required products modify the option in the **Backorder** column from **Use Config Settings** that is selected by default to the one you need.



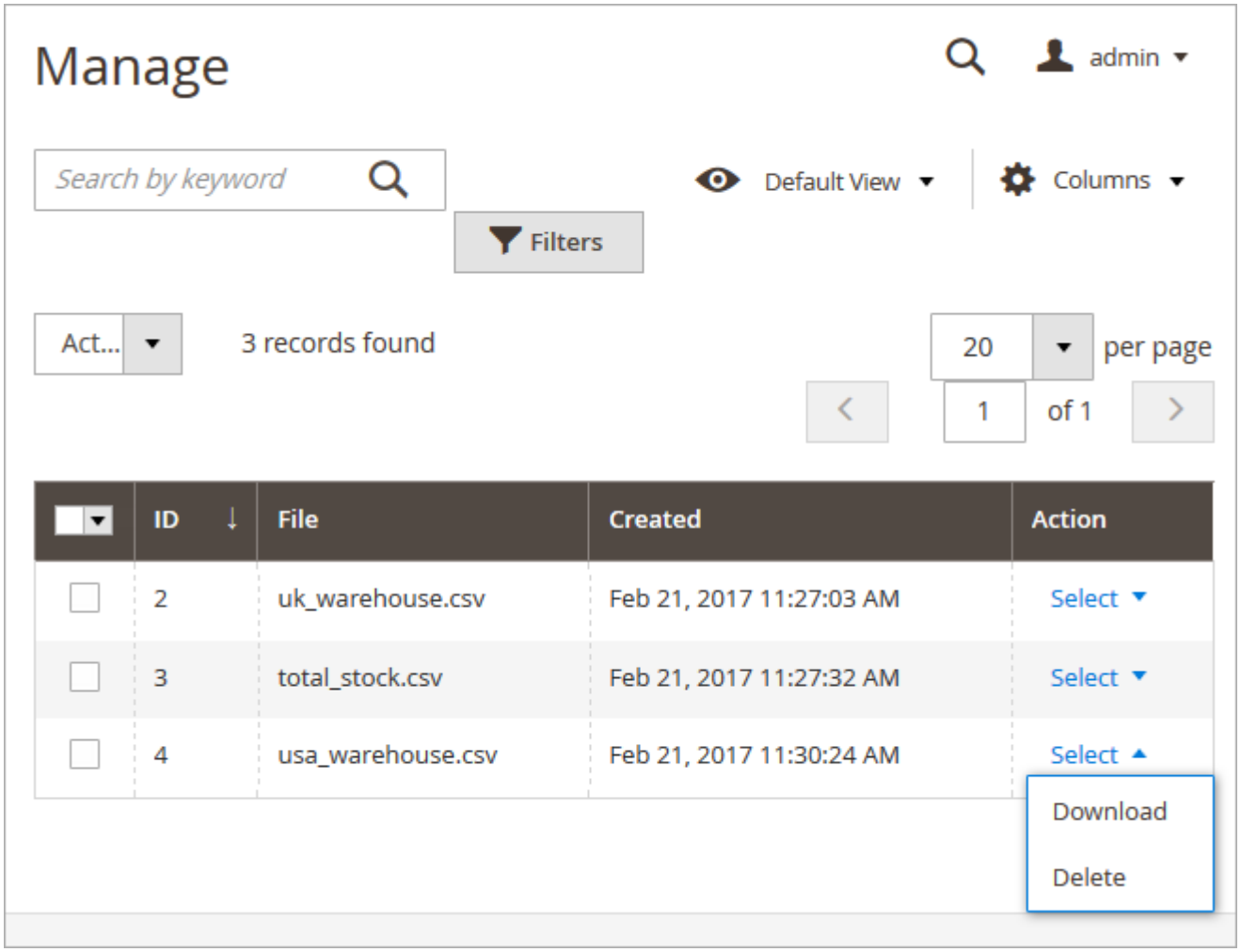
- You can disable backorders (No backorders) or enable them (Allow Qty Below 0, Allow Qty Below 0 and Notify Customer. Set the option back to **Use Config Settings** if you need the product to utilize global backorders settings.



- Repeat this operation for all required products in each warehouse.

Export History

To open the Export History grid please go to **Products → Inventory → Export History**.



Export History Grid - On the grid, you will see all stock data export attempts. It's possible to download export data to a CSV file or delete them.

Below you'll see the **example of an exported CSV file**.

```
sku,code,qty
24-MB04,default,13.0000
24-MB04,UK,2.0000
24-MB04,USA,5.0000
24-MB04,Australia,6.0000
```

Import Stock

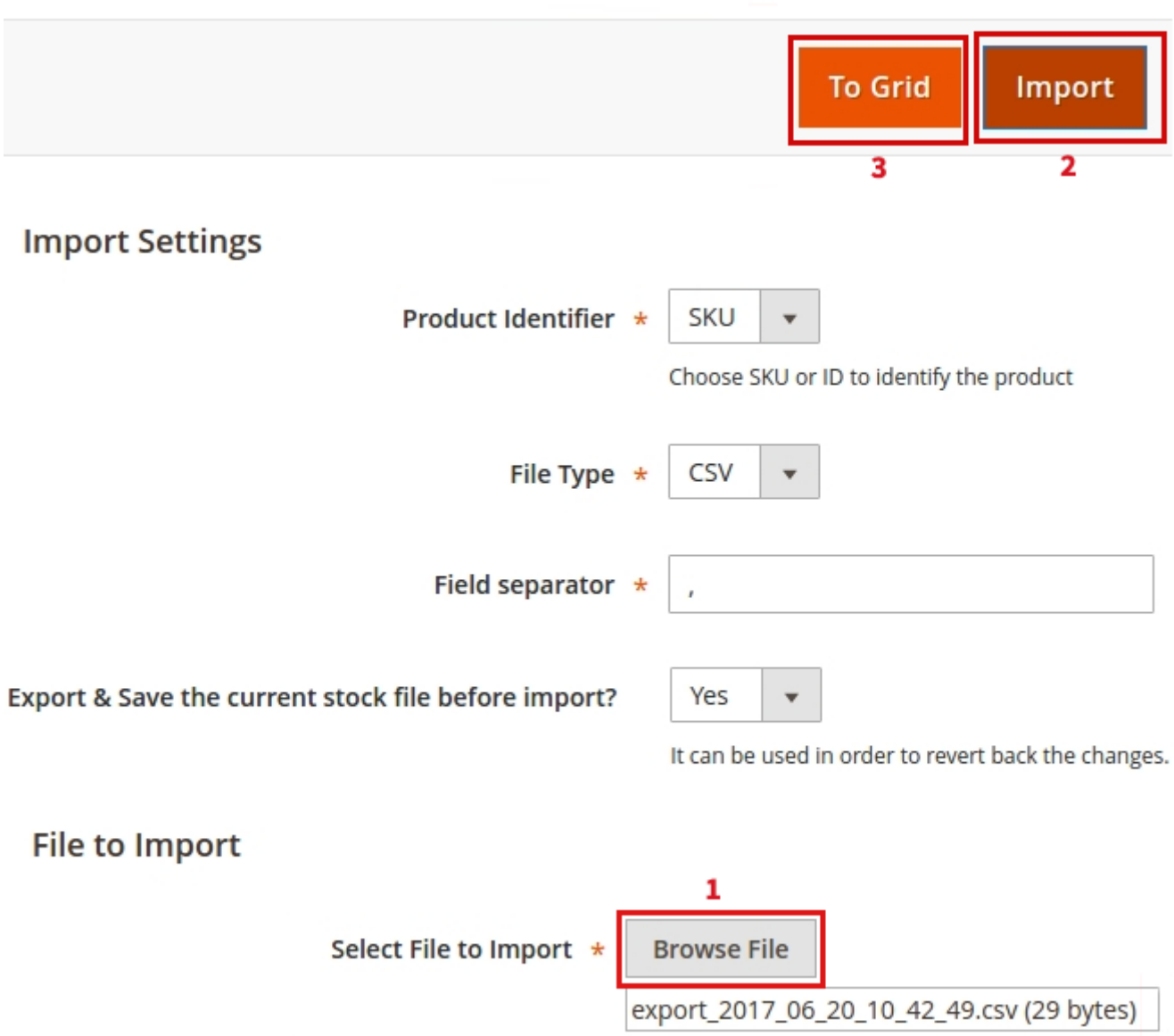
To import files with stock data go to **Products → Inventory → Import Stock**.

Import Stock - specify product identifier (SKU or ID) and import stock data using a CSV or XML file.

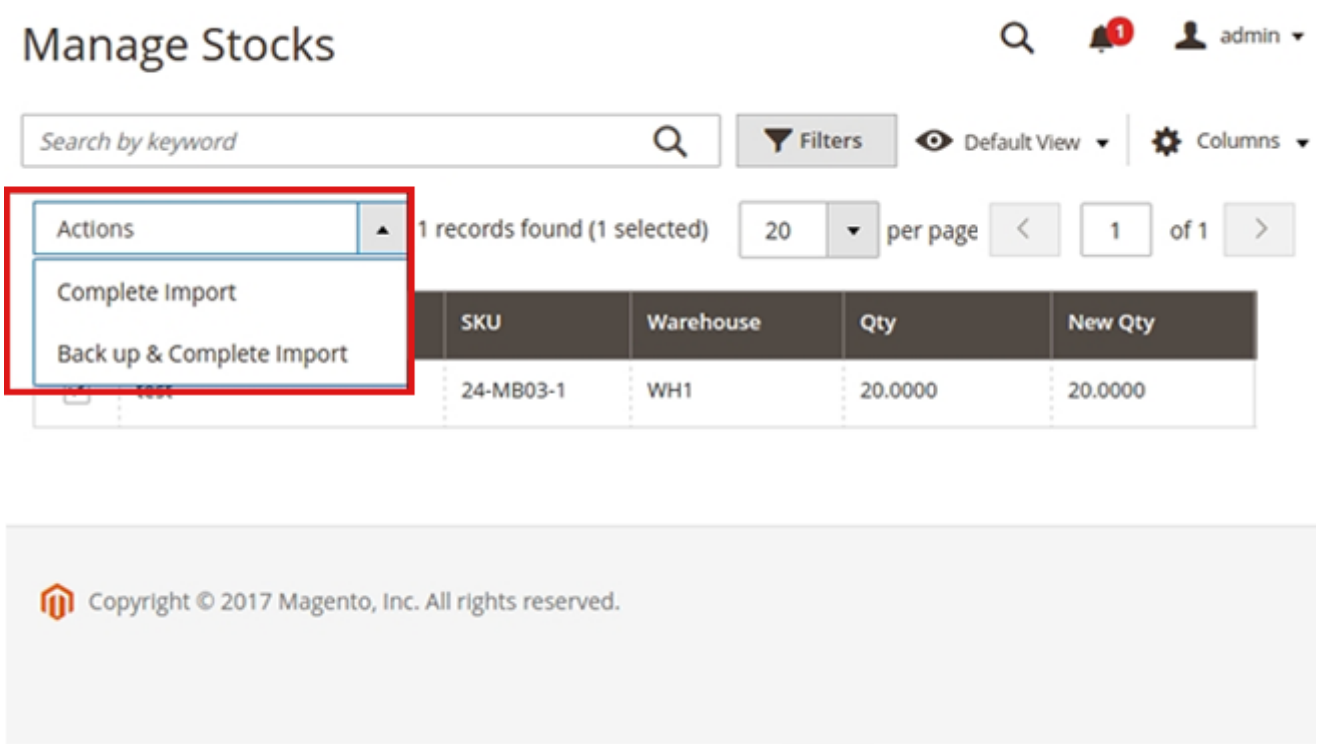
To import stock data you need to follow the next steps:

1) First of all, browse and **upload a file** in CSV or XML format

2) Press **“Import”**, and then a **“To grid”** button will appear on the screen. Click on it to continue.



3) After you make some changes on the grid, choose the items you need and on the tab **“Actions”** select one variant.



For stock data import you can use the same format as shown in the previous block example.

You can also change the file in the following way:

- in the 1st column you can use either SKU or ID (sku/id);
- in the **quantity column** you can insert + or -;
- in the **code column** please specify the code of your warehouse (except for the default one).

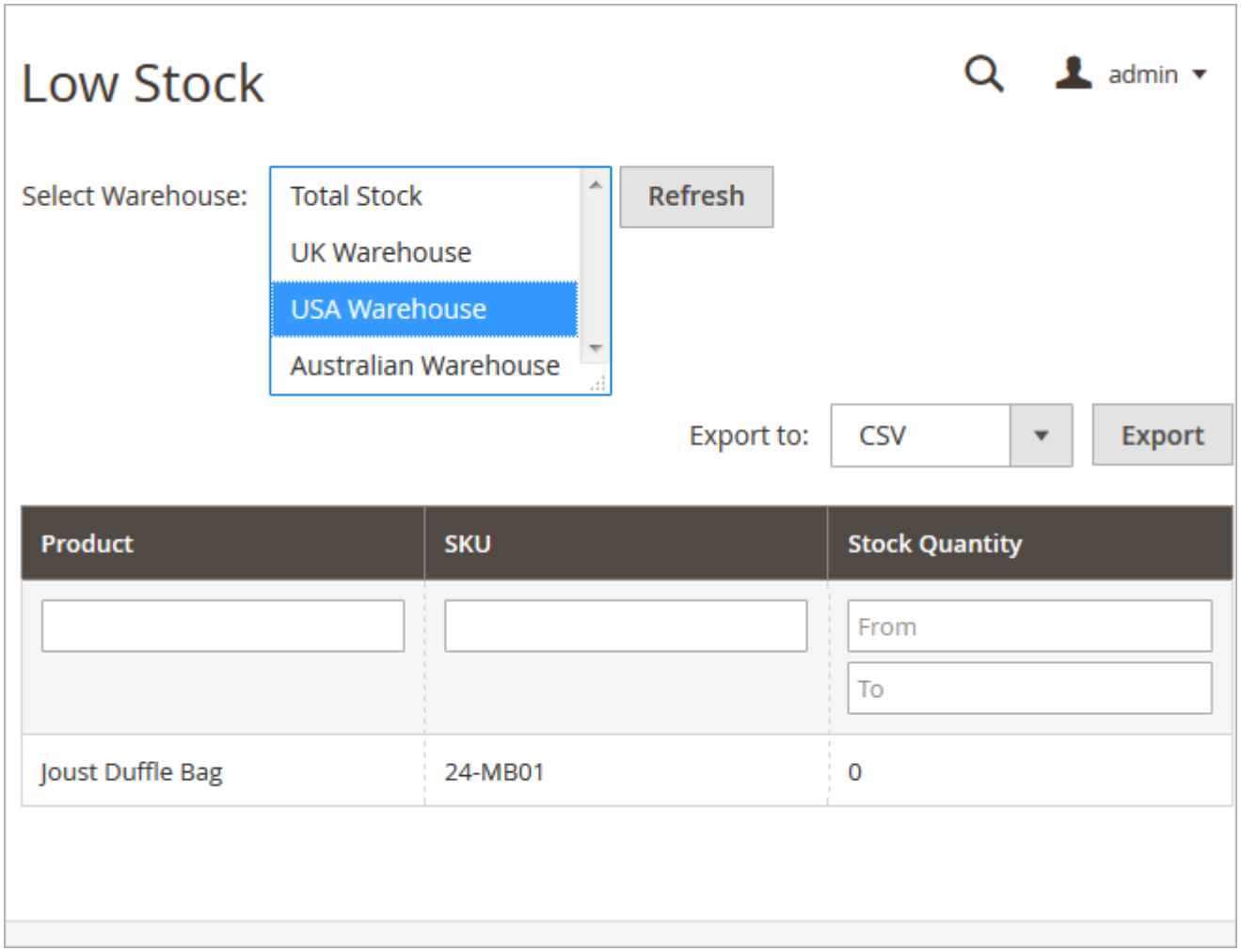
Each product has one field for specifying a shelf location, but you can insert here several values separated by a comma or by space.

```
id,code,qty
95,default,13.0000
95,UK,2.0000
95,USA,5.0000
95,Australia,6.0000
```

In our example, we've used product ID instead of SKU.

Low Stock Report

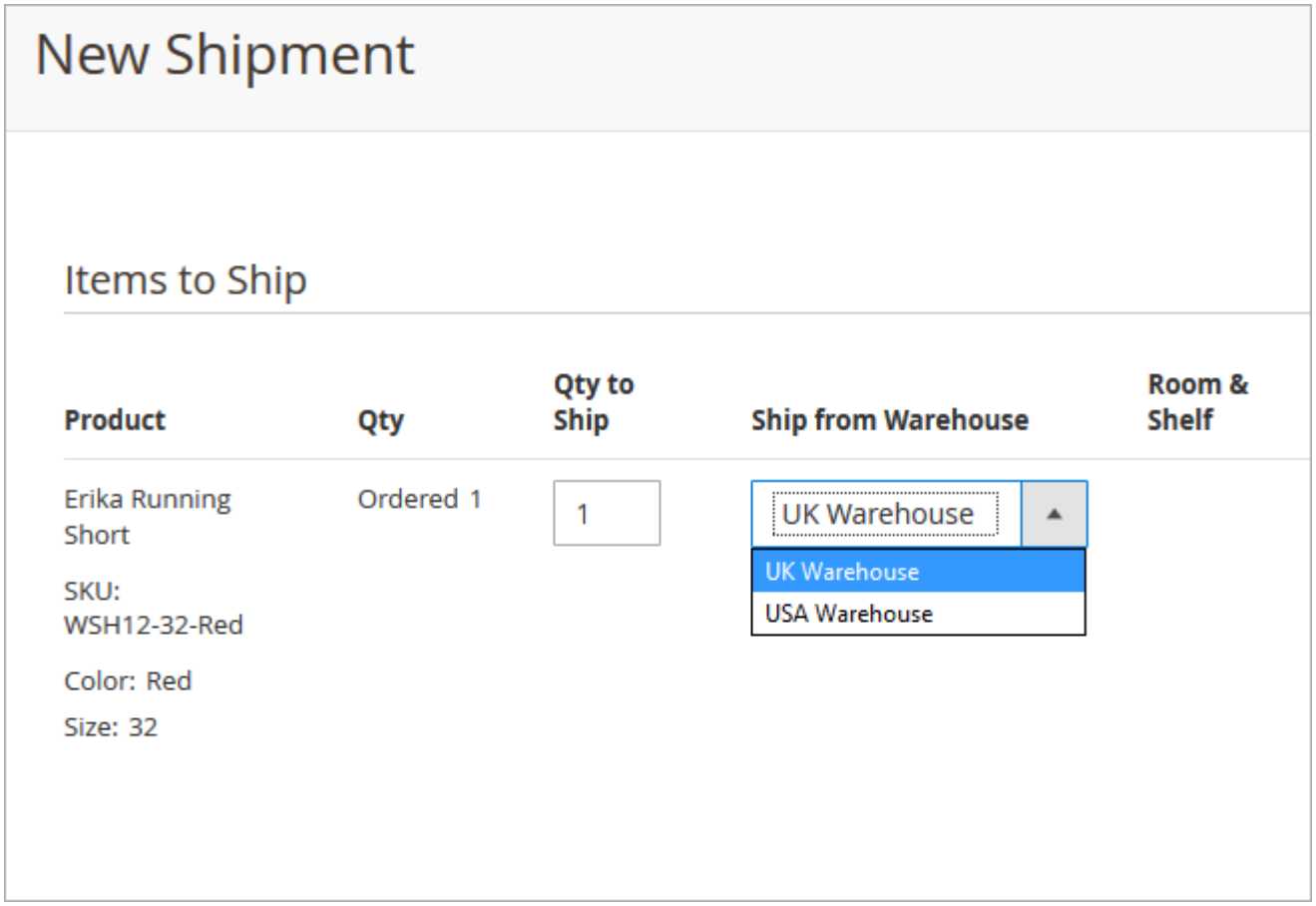
To view and export a low-stock report go to **Reports → Warehouse Low Stock**.



Low Stock Report Grid - Filter report data by a warehouse or view details for all warehouses at a time. Export report data to a CSV or XML file.

Order Shipment

When an order contains the item assigned to multiple warehouses, the most appropriate warehouse will be selected according to the criteria you specified in general stock settings in the 'Choose warehouse based on' setting.



Ship from Warehouses - the most appropriate warehouse will be selected automatically according to the above-mentioned settings, but you can select a warehouse manually when creating a shipment.

Warehouse Selection Logic

- When the order is placed, the extension uses the Available Qty of the product in the Total Stock (default). The extension assigns a warehouse to every order item. You can change the warehouses associated to order items during the order shipment.
- If the item is available in several warehouses, the module selects the warehouse according to specified parameters (nearest warehouse, store view of the order, warehouse priority, customer group):

1. The extension starts filtering warehouses by the first criteria.

2. If more than one warehouse is determined, the second parameter is used. And so on until the warehouse is determined.

3. If after filtering by all specified criteria there are still 2 or more warehouses, the module will select the one with the highest stock level.

4. In case the most appropriate warehouse doesn't contain the necessary amount of ordered items,

the missing amount will be taken from another warehouse.

Update and retrieve stock data via Magento SOAP API and REST API

To send or retrieve data via REST API or SOAP you need to get the **Oath token**.

To get the key, execute the following command in the ssh console:

```
 curl -X POST "http://magentohost/index.php/rest/V1/integration/admin/token"
-H "Content-Type:application/json" -d '{"username":"user",
"password":"password"}'
```

! Instead of 'magentohost' specify your URL, instead of 'user' indicate your admin login and use an admin password instead of 'password'.

After that you'll get **the key (e.g. $key)**.

REST API

Service: amastyMultiInventoryWarehouseItemRepositoryV1

/V1/amastymultisetstock: - Send data in the json format to add them to stock.

/V1/amastymultigetstocks/{id}/ - Get information about stocks where the product is stored. For {id} specify the corresponding product id.

/V1/amastymultigetproducts/{code}/ - Retrieve the list of products (+ products quantity) which are stored in the warehouse.

1. Send data via SSH

1.1 With the URL

```
 curl -X POST --header "Content-Type: application/json" --header "Accept:
application/json" --header "Authorization: Bearer $key" -d "{
  \"warehouseItem\": {
    \"product_id\": 11,
    \"qty\": 10,
    \"room_shelf\": \"BB\",
    \"warehouse_id\": 5
  }
}" "http://magentohost/rest/default/V1/amastymultisetstock"
```

!Insert your own URL instead of 'magentohost' and use the Oath token you've got instead of our example key ($key).

```
 curl -X GET --header "Accept: application/json" --header "Authorization:
Bearer $key" "http://magentohost/rest/default/V1/amastymultigetstocks/12/"
```

! '12'– This is an example product id. Here you should indicate the id of the product you need.

In response we'll get the following data:


```
[
  {
    "product_id": 12,
    "qty": 10,
    "available_qty": 9,
    "ship_qty": 1,
    "room_shelf": null,
    "id": 6267,
    "warehouse_id": 1
  },
  {
    "product_id": 12,
    "qty": 10,
    "available_qty": 9,
    "ship_qty": 1,
    "room_shelf": "",
    "id": 8388,
    "warehouse_id": 3
  },
  {
    "product_id": 12,
    "qty": 10,
    "available_qty": 10,
    "ship_qty": 0,
    "room_shelf": "BB",
    "id": 8398,
    "warehouse_id": 5
  }
]
```


```
 curl -X GET --header "Accept: application/json" --header "Authorization:
Bearer $key"
"http://magentohost/rest/default/V1/amastymultigetproducts/warehouse1/"
```

! 'warehouse1'- This is an example of a warehouse code. Please, specify the code of the warehouse you need.

In response we get the list of products:


```
[
  {
```

```
    "product_id": 2042,
    "qty": 30,
    "available_qty": 22,
    "ship_qty": 8,
    "room_shelf": "BB",
    "id": 8303,
    "warehouse_id": 3
  },
  {
    "product_id": 2043,
    "qty": 18,
    "available_qty": 18,
    "ship_qty": 0,
    "room_shelf": "AA",
    "id": 8304,
    "warehouse_id": 3
  },
  {
    "product_id": 1,
    "qty": 10,
    "available_qty": 10,
    "ship_qty": 0,
    "room_shelf": "22",
    "id": 8306,
    "warehouse_id": 3
  },
  {
    "product_id": 13,
    "qty": 20,
    "available_qty": 20,
    "ship_qty": 0,
    "room_shelf": "AA",
    "id": 8314,
    "warehouse_id": 3
  }
]
```

2. Send data via PHP

2.1 With the URL

```
    $token = $key;
    $authorization = "Authorization: Bearer " . $token;
    $domain = 'http://magentohost/rest/default/V1/amastymultisetstock';
    $post_data = [
      "warehouseItem" => [
          "product_id" => 11,
```

```
            "qty" => 10,
            "room_shelf" => "BB",
            "warehouse_id" => 5
          ]
        ];
      $curlInit = curl_init($domain);

      curl_setopt($curlInit, CURLOPT_HTTPHEADER, ['Content-Type:
application/json' , 'Accept: application/json', $authorization ]);
      curl_setopt($curlInit,CURLOPT_CONNECTTIMEOUT,30);
      curl_setopt($curlInit, CURLOPT_POST, true);
      curl_setopt($curlInit, CURLOPT_POSTFIELDS, json_encode($post_data));
      curl_setopt($curlInit, CURLOPT_RETURNTRANSFER, true);

      $response = curl_exec($curlInit);

      curl_close($curlInit);
      echo $response;
```

! Insert your own URL instead of 'magentohost and use the Oath token you've got instead of our example key ($key). The data is transferred in array. Place the data in json_encode() and send.

Execute the following command to receive data on GET functions:

```
$token = $key;
    $authorization = "Authorization: Bearer " . $token;
    $domain = 'http://magentohost/rest/default/V1/amastymultigetstocks/12/';
    $curlInit = curl_init($domain);
    curl_setopt($curlInit, CURLOPT_HTTPHEADER, array('Content-Type:
application/json' , $authorization ));
    curl_setopt($curlInit, CURLOPT_CONNECTTIMEOUT,30);
    curl_setopt($curlInit, CURLOPT_CUSTOMREQUEST, "GET");
    curl_setopt($curlInit, CURLOPT_RETURNTRANSFER, true);

    $response = curl_exec($curlInit);

    curl_close($curlInit);
    echo $response;
```

2.2 Zend Framework

It’s also possible to send data via Zend Framework used in Magento:

```
      $token = $key;
      $httpHeaders = new \Zend\Http\Headers();
      $httpHeaders->addHeaders([
         'Authorization' => 'Bearer ' . $token,
         'Accept' => 'application/json',
         'Content-Type' => 'application/json'
```

```
    ]);

    $request = new \Zend\Http\Request();
    $request->setHeaders($httpHeaders);
$request->setUri('http://magentohost/rest/default/V1/amastymultisetstock');
    $request->setMethod(\Zend\Http\Request::METHOD_POST);
    $post_data = [
       "warehouseItem" => [
           "product_id" => 11,
           "qty" => 10,
           "room_shelf" => "BB",
           "warehouse_id" => 5
      ]
   ];
   $params = new \Zend\Stdlib\Parameters($post_data);
   $request->setQuery($params);

   $client = new \Zend\Http\Client();
   $options = [
       'adapter'   => 'Zend\Http\Client\Adapter\Curl',
       'curloptions' => [CURLOPT_FOLLOWLOCATION => true],
       'maxredirects' => 0,
       'timeout' => 30
   ];
   $client->setOptions($options);

   $response = $client->send($request);
   echo $response->getContent();
```

!Insert your own URL instead of ‘magentohost’ and use the Oath token you've got instead of our example key ($key).

Execute the following command to receive data on GET functions:

```
   $token = $key;
   $httpHeaders = new \Zend\Http\Headers();
   $httpHeaders->addHeaders([
       'Authorization' => 'Bearer ' . $token,
       'Accept' => 'application/json',
       'Content-Type' => 'application/json'
   ]);

   $request = new \Zend\Http\Request();
   $request->setHeaders($httpHeaders);
$request->setUri('http://mli213.chernov.p70m.sty/rest/default/V1/amastymulti
getstocks/12/');
   $request->setMethod(\Zend\Http\Request::METHOD_GET);
```

```
    $client = new \Zend\Http\Client();
    $options = [
        'adapter'   => 'Zend\Http\Client\Adapter\Curl',
        'curloptions' => [CURLOPT_FOLLOWLOCATION => true],
        'maxredirects' => 0,
        'timeout' => 30
    ];
    $client->setOptions($options);

    $response = $client->send($request);
    echo $response->getContent();
```

SOAP

Service: amastyMultiInventoryWarehouseItemRepositoryV1

amastyMultiInventoryWarehouseItemRepositoryV1addStock - Send data in the json format to add them to stock.

amastyMultiInventoryWarehouseItemRepositoryV1getStocks - Get information about stocks where the product is stored. For {id} specify the corresponding product id.

amastyMultiInventoryWarehouseItemRepositoryV1getProducts - Retrieve the list of products (+quantity) which are stored in the warehouse.

1. Send data via PHP

Send data for stock update

```
$url =
'http://magentohost/index.php/soap/all?wsdl&services=amastyMultiInventoryWar
ehouseItemRepositoryV1';
   $params = [
        'http'=>[
            'header' => 'Authorization: Bearer ' . $key
        ]
   ];
   $context = stream_context_create($params);
   $client = new SoapClient($url,  ['soap_version'=> SOAP_1_2,
'stream_context'=>$context]);

   $post_data = [
        'warehouseItem' => [
            "id" => null,
            "productId" => 11,
            "qty" => 30,
            "availableQty" => 0,
            "shipQty" => 20,
            "roomShelf" => "BB",
```

```
            "warehouseId" => 5
        ]
    ];

    $soapResponse =
$client->amastyMultiInventoryWarehouseItemRepositoryV1addStock($post_data);

    echo json_encode($soapResponse);
```

! Insert your own URL instead of 'magentohost' and use the Oath token you've got instead of our example key ($key).

Receive data via Get

```
$url =
'http://magentohost/index.php/soap/all?wsdl&services=amastyMultiInventoryWar
ehouseItemRepositoryV1';
    $params = [
        'http'=>[
            'header' => 'Authorization: Bearer ' . $key
        ]
    ];
    $context = stream_context_create($params);
    $client = new SoapClient($url,  ['soap_version'=> SOAP_1_2,
'stream_context'=>$context]);
    $soapResponse =
$client->amastyMultiInventoryWarehouseItemRepositoryV1getStocks(
     ['id'           => 12]);

    echo json_encode($soapResponse);
```

Use SKU in API

You can use either product SKU or the product ID & warehouse code or warehouse ID
Find out how to install the **Multi Warehouse Inventory** extension for Magento 2 via Composer.

Rate the user guide
★★★★☆ from 7 votes (Details)
○ ○ ○ ○ ○ Rate
★☆☆☆☆ 0 visitor votes
★★☆☆☆ 1 visitor votes
★★★☆☆ 0 visitor votes
★★★★☆ 1 visitor votes
★★★★★ 3 visitor votes

From:
https://amasty.com/docs/ - **Amasty Extensions FAQ**

Permanent link:
https://amasty.com/docs/doku.php?id=magento_2:multi_warehouse_inventory

Last update: **2017/11/15 11:01**