Miinto order sync manual

⚠ The Miinto order synchronization is currently supported for **Shopify, Magento**, **Magento 2**, **WooCommerce**, **Lightspeed**, **Shoplo**, **Prestashop** , **BigCommerce** and **CCV Shop** platforms.

Update your Koongo plan

Prior the Miinto order sync setup, you need to upgrade your plan and add a **connection**. For more details, please check My Plan & Pricing manual.

⚠ This step is not applied to the **CCV Shop** platform.

Order sync setup

To activate the Miinto order synchronization, please follow the steps below:

Add connection

Click **Add Connection** button.

Select the Miinto connection

Select the Miinto connection from the list

Fill user details

Fill your Miinto account **username and password**, set **Test Mode** to **"No"**. Click **Get Available Stores** button.

Select store

Select **Store Id** and set **Activate Order Synchronization** to **ACTIVE**. Save the settings.

Sync your orders

Click **Synchronize Orders** button

Orders configuration

You can modify various parameters in the **Orders Config** section

Activate Order Synchronization

You can set the connection to **Active** or **Disabled**.

Order Synchronization Level

The Order Synchronization Level option can be set to **Orders&Stock** or **Stock only**.

Accept Orders Automatically

The Accept Orders Automatically option can be set to **Yes.**

> ⓘ **Accept Orders Automatically - Yes**
>
> Koongo can automatically decide whether Miinto order should be accepted or not. Acceptance is based on the merchant stock and price of certain order items.
>
> The order is accepted automatically if the product is in stock and the order price equals or is higher than the reference Price Attribute - **Price final include tax** or **Price original include tax** can be used.

By default, the Accept Orders Automatically option is set to **No**. Each order needs to be accepted manually in your Miinto or Koongo account.

Auto Accept Orders - Price Attribute

Auto accept order functionality price reference can be set either to **Price final include tax** or **Price original include tax**. By default **Price final include tax** attribute is used.

Auto Accept Orders - Price Attribute Threshold

Threshold which is applied to price attribute before comparison with order item's price (price attribute is multiplied by threshold). Recommended value is 0.97, which means that order is accepted if order item price is equal or higher than 97% of product price.

Shipping Provider

You can define the shipping service default value. The default value will be used in case the order shipping service defined in the store doesn't match Miinto allowed providers.
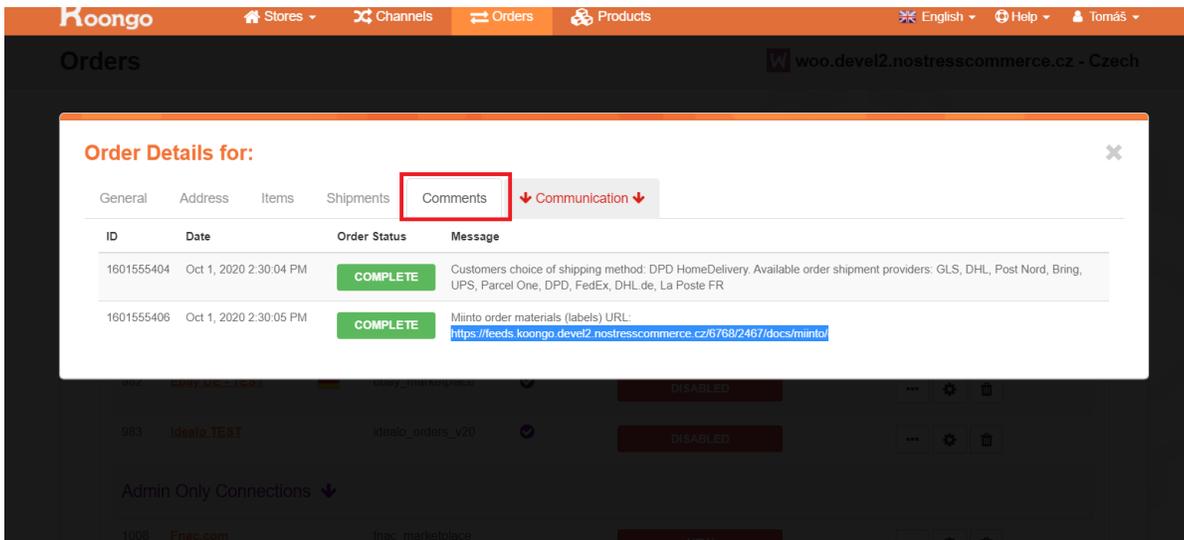
Product Identification Attribute

You can select attribute for product identification on Miinto. By default id attribute is used.

Order details

You can check each order details.

Miinto Labels Transfer

In the Miinto order Comment tab, there is a link to the order labels in PDF. The file is stored at our server, but only users who know the link can access it.

Miinto Return merchandise authorization (RMA)

> ⚠️ **Miinto RMA availability**
>
> Miinto Return merchandise authorization (RMA) is currently available for all order sync supported e-commerce platforms. The Miinto order synchronization needs to be activated to process RMA.
>
> The **refund synchronization is available only for Shopify users**. The other e-commerce platforms will be added later this year.

Return merchandise authorization (RMA) is a part of the process of returning a product to receive a refund, replacement, or repair during the product's warranty period.

Koongo supports Miinto **RMA functionality** (Return merchandise authorization) as part of the Miinto order management. This functionality allows merchants to process Miinto RMA in Koongo.

Koongo RMA supported features

- RMA management at Koongo
    - Check RMAs at particular orders
    - Accept RMA
    - Reject RMA with the definition of rejection reason
- Once RMA is accepted at Koongo or Miinto, then a relevant refund action is translated from Miinto to the store.
- Refund syncing - Once the merchant refunds order items at the store, then it will be transferred into Miinto and a new RMA Request is created. (**Available only for Shopify platform**)

Miinto RMA and RMA request processing

Scenario 1 - Buyer returns the product, no RMA created in Miinto

- The buyer sends back the product to the merchant without creating RMA in Miinto
- The merchant check the returned product
- The merchant creates an RMA Request in Miinto
    - The Shopify users can create a product refund in Shopify. Then, the RMA Request is created in Miinto automatically
- The refund itself will be performed by Miinto marketplace

Scenario 2 - Buyer creates RMA, merchant accepts RMA

- The buyer creates an RMA in Miinto and sends the product back to the merchant
- The merchant check the returned product
- The merchant accepts the RMA in Miinto or in Koongo

- The refund itself will be performed by Miinto marketplace. For Shopify platform, the refund is sent to merchant store automatically.
- Other than Shopify platform users need to insert the return manually.
- Email notification is sent to the buyer - 'Your return has been accepted'

Scenario 3 - Buyer creates RMA, merchant declines RMA

- The buyer creates an RMA in Miinto and sends back the product to the merchant
- The merchant check the returned product
- The merchant decline the RMA in Miinto or in Koongo. And provide RMA with declined reason to Miinto
- The RMA is declined in Miinto system
- The merchant sends the product back to the customer
- Email notification is sent to the buyer - 'Your return has been declined'

Miinto RMA processing flow - diagram

Miinto RMA processing in Koongo

Pending RMA

An RMA is created in Miinto and imported to Koongo.

You can **accept** or **decline** the RMA. You should fill the accept or decline reason description.

Accepted/Declined RMA

The RMA is marked as **Accepted/Declined.**

Refund (Shopify supported only)

The refund is transferred to the store automatically. **This functionality is available only for Shopify platform.**