Odoo 2 Magento Connector - User Manual

POST CH AG

EXECUTIVE SUMMARY | MIHAI CIMPEANU | EPOINT | DECEMBER 2017 | TIMISOARA | ROMANIA

Table of contents

1. ABOUT

- Swiss Post is consolidating its core business (logistics) by offering integrated software solutions. For a better service it has upgraded its ERP solution, on an Odoo open source framework and in order to offer an easier onboarding for e-commerce customers, we developed an open source connector between Odoo and Magento.
- Potential customers can use the free Magento Commerce extension to integrate the front store with the ERP.
- The Magento Commerce Extension will exchange products, stocks, orders, customers and logistics information with the Odoo ERP.

2. INSTALLATION

- This guide will help you to install the Magento Commerce Extension that makes the connection between the Magento ecommerce store and Odoo ERP.

2.1. REQUIREMENTS

- Magento CE: 2.2+
- Php 7.0.2, 7.0.4, 7.0.6-7.0.x, 7.1.x
- Php libraries: php json, php curl

2.2. INSTALLATION OF THE MODULE

Step 1: Download the extension: https://github.com/epointwebsolutions/Post-CH-Odoo-ERP-Magento2-Connector
Step 2: Unzip the files
Step 3: Copy the files to {Magento root}/app/code/
Step 4: Flush cache
Step 5: Setup the module by navigating to {Magento root} and use the following commands:

- ./bin/magento setup:upgrade
- ./bin/magento cache:flush

3. CONFIGURATION OF THE MODULE

- After the module setup, in order for Magento to connect with the Odoo ERP, there are a couple of configurations that needs to be made.
- To access the configuration sections go to **Stores > Configuration > Epoint Swisspost API.**
- In the following sub-chapters, each configuration section will be discuss in details.

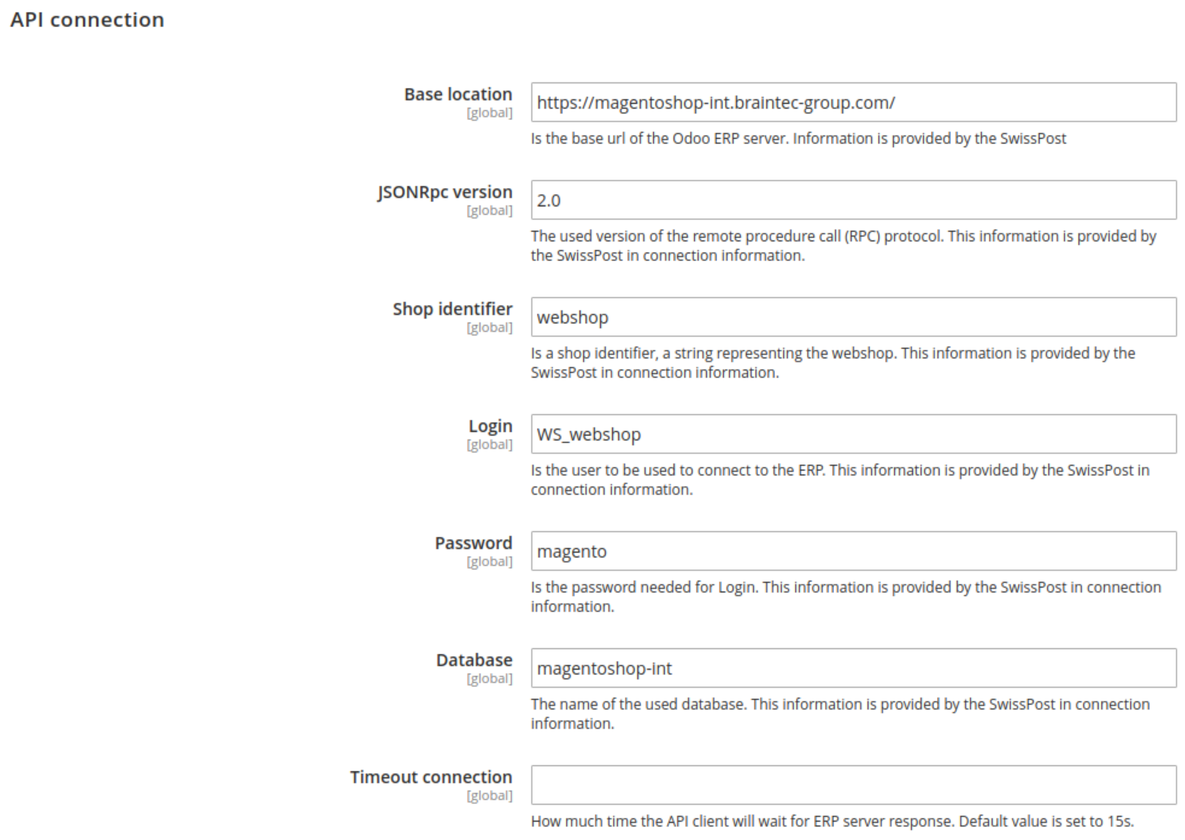
3.1. SECTION > API CONNECTION

This section contain the configuration needed for module to connect to Odoo ERP and for logging management.

a. API CONNECTION

To connect to the Odoo ERP, the first thing that we need to do is to configure the connection credentials.

- **Base Location**
  - Is the base url of the Odoo ERP server
  - This information is provided by SwissPost
- **JSONRpc version**
  - The used version of the remote procedure call (RPC) protocol, in this case 2.0
  - The information is provided by SwissPost
- **Shop identifier**
  - Is a shop identifier, a string representing the web shop
  - The information is provided by SwissPost
- **Login**
  - Is the user to be used to connect to the ERP
  - The information is provided by SwissPost
- **Password**
  - Is the password needed for **Login**
  - The information is provided by SwissPost
- **Database**
  - The name of the used database
  - The information is provided by SwissPost
- **Timeout connection**
  - How much time the API client will wait for server response
  - Default value is set to 15s



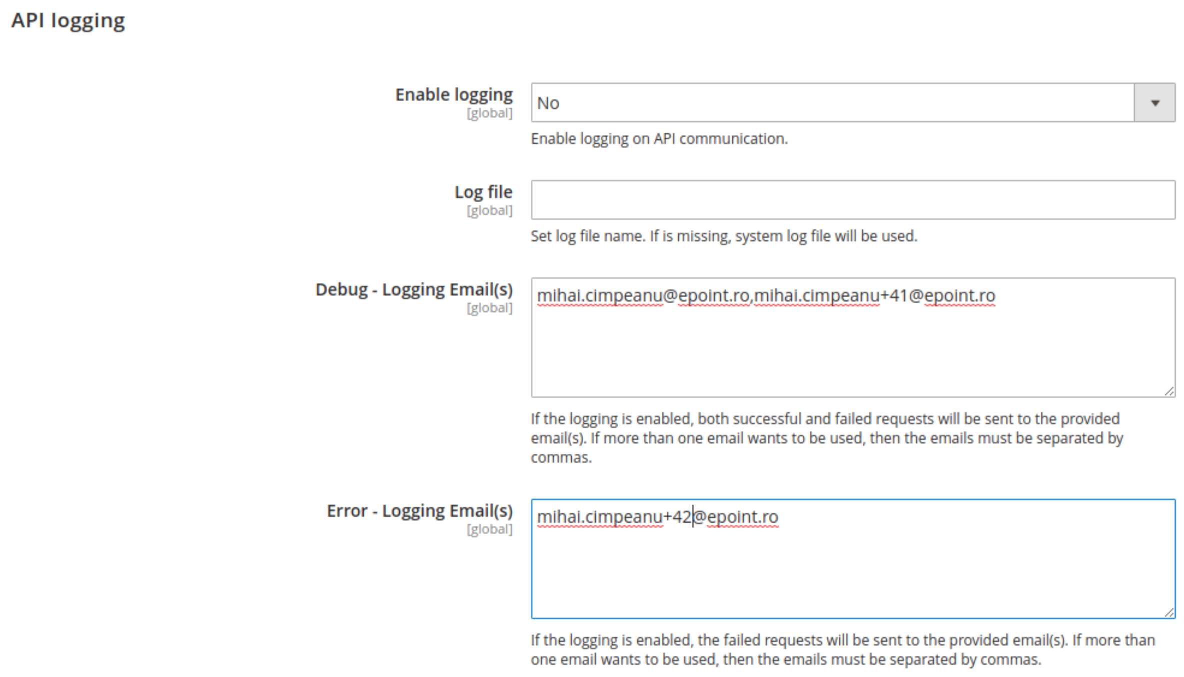
API connection

Base location [global]	https://magentoshop-int.braintec-group.com/ Is the base url of the Odoo ERP server. Information is provided by the SwissPost
JSONRpc version [global]	2.0 The used version of the remote procedure call (RPC) protocol. This information is provided by the SwissPost in connection information.
Shop identifier [global]	webshop Is a shop identifier, a string representing the webshop. This information is provided by the SwissPost in connection information.
Login [global]	WS_webshop Is the user to be used to connect to the ERP. This information is provided by the SwissPost in connection information.
Password [global]	magento Is the password needed for Login. This information is provided by the SwissPost in connection information.
Database [global]	magentoshop-int The name of the used database. This information is provided by the SwissPost in connection information.
Timeout connection [global]	 How much time the API client will wait for ERP server response. Default value is set to 15s.

b. API LOGGING

This tab contains the configuration of logging system.

- **Enable logging**
  - Enable or disable the logging on API communication
- **Log file**
  - The name of the file where the logs can be found
  - If no file name has been provided, the system log file will be used
- **Debug - Logging Email(s)**
  - Must contain a list of emails separated by commas
  - If the logging is enabled, both successful and failed requests will be sent to the provided emails
- **Error - Logging Email(s)**
  - Must contain a list of emails separated by commas
  - If the logging is enabled, the failed requests will be sent to the provided emails


API logging

Enable logging [global]: No
Enable logging on API communication.

Log file [global]:
Set log file name. If is missing, system log file will be used.

Debug - Logging Email(s) [global]: mihai.cimpeanu@epoint.ro,mihai.cimpeanu+41@epoint.ro
If the logging is enabled, both successful and failed requests will be sent to the provided email(s). If more than one email wants to be used, then the emails must be separated by commas.

Error - Logging Email(s) [global]: mihai.cimpeanu+42@epoint.ro
If the logging is enabled, the failed requests will be sent to the provided email(s). If more than one email wants to be used, then the emails must be separated by commas.

3.2 SECTION > SALES

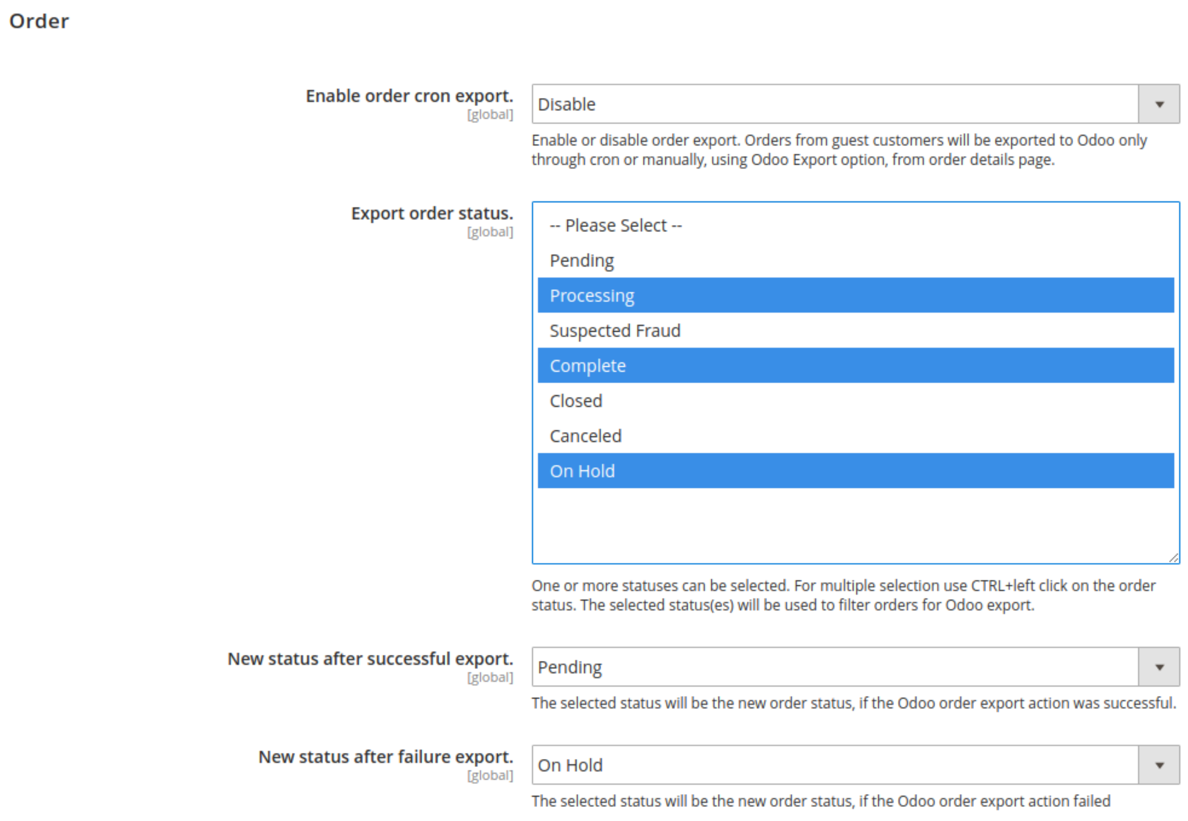
This section contain the configuration needed for:

- Exporting orders, new accounts, coupons as gift cards
- Importing order payment and transfer status
- Mapping payment and shipping methods, customer groups and tax classes between Odoo ERP and Magento
- Printing order invoices or shipping reports from Odoo ERP

a. ORDER

- **Enable order cron export**
  - Enable or disable order cron export
  - Orders from guest customers will be exported to Odoo only through cron or manually using **Odoo Export** option from order details page

- **Export order status**
    - The selected status(es) will be used to filter orders for Odoo export
    - One or more statuses can be selected
    - For multiple selection use CTRL+ left click on the order status
- **New status after successful export**
    - The selected status will be the new order status, if the Odoo order export action was successful
- **New status after failure export**
    - The selected status will be the new order status, if the Odoo order export action failed


Order

Enable order cron export. [global]: Disable

Enable or disable order export. Orders from guest customers will be exported to Odoo only through cron or manually, using Odoo Export option, from order details page.

Export order status. [global]:
- -- Please Select --
- Pending
- Processing (selected)
- Suspected Fraud
- Complete (selected)
- Closed
- Canceled
- On Hold (selected)

One or more statuses can be selected. For multiple selection use CTRL+left click on the order status. The selected status(es) will be used to filter orders for Odoo export.

New status after successful export. [global]: Pending

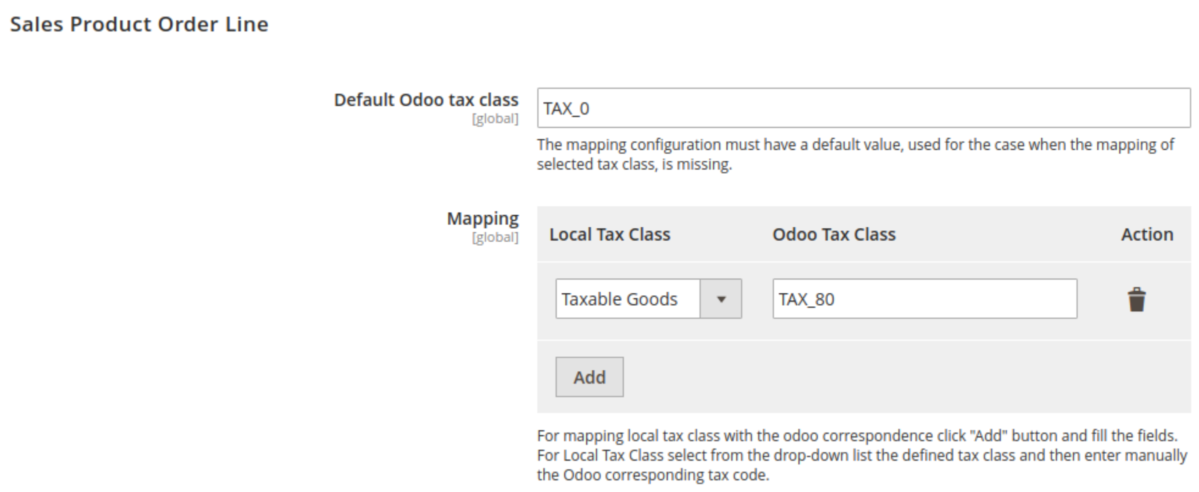
The selected status will be the new order status, if the Odoo order export action was successful.

New status after failure export. [global]: On Hold

The selected status will be the new order status, if the Odoo order export action failed

b. SALES PRODUCT ORDER LINE

- **Default Odoo tax class**
    - The mapping configuration must have a default value, used for the case when the mapping of selected tax class, is missing.
- **Mapping**
    - Will map the local tax class with Odoo correspondence
    - For mapping local tax class with the odoo correspondence, click "Add" button and fill the fields.
    - For Local Tax Class, select from the drop-down list the defined tax class and then, enter manually, the Odoo corresponding tax code.



Sales Product Order Line

Default Odoo tax class [global]: TAX_0

The mapping configuration must have a default value, used for the case when the mapping of selected tax class, is missing.

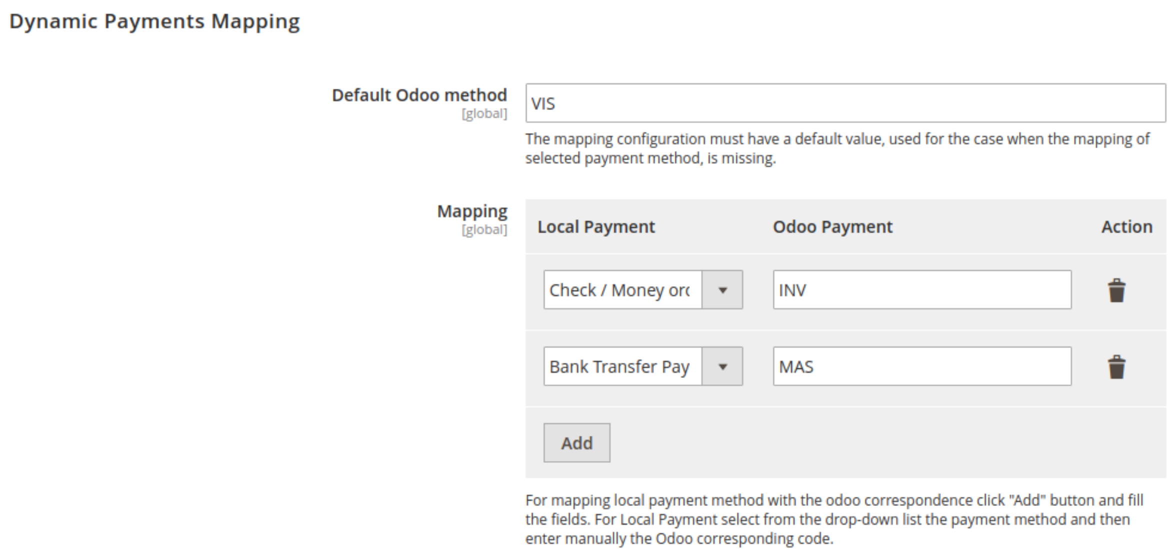
Mapping [global]

Local Tax Class	Odoo Tax Class	Action
Taxable Goods	TAX_80	

For mapping local tax class with the odoo correspondence click "Add" button and fill the fields. For Local Tax Class select from the drop-down list the defined tax class and then enter manually the Odoo corresponding tax code.

c. DYNAMIC PAYMENTS MAPPING

- **Default Odoo method**
  - The mapping configuration must have a default value, used for the case when the mapping of selected payment method, is missing.
- **Mapping**
  - Will map the local payment method with Odoo correspondence
  - For mapping local payment method with the Odoo correspondence, click "Add" button and fill the fields.
  - For Local Payment select from the drop-down list the payment method and then, enter manually, the Odoo corresponding code.



Dynamic Payments Mapping

Default Odoo method [global]: VIS

The mapping configuration must have a default value, used for the case when the mapping of selected payment method, is missing.

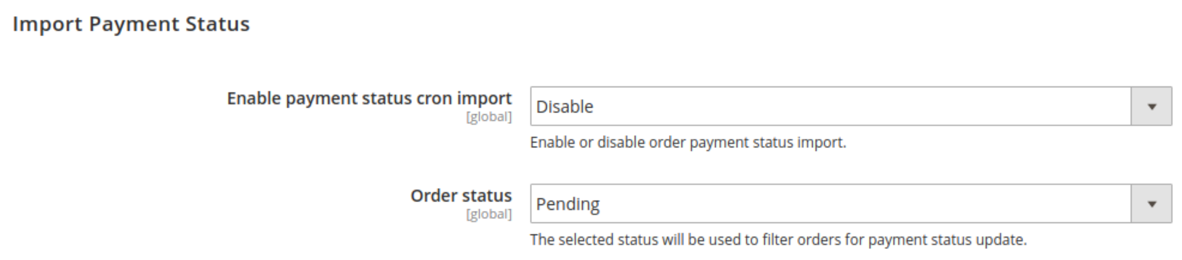
Mapping [global]

Local Payment	Odoo Payment	Action
Check / Money orc	INV	
Bank Transfer Pay	MAS	

For mapping local payment method with the odoo correspondence click "Add" button and fill the fields. For Local Payment select from the drop-down list the payment method and then enter manually the Odoo corresponding code.

d. IMPORT PAYMENT STATUS

- **Enable payment status cron import**
  - Enable or disable cron import of order payment status
- **Order status**
  - The selected status will be the accepted order status for cron import of the payment status


Import Payment Status

Enable payment status cron import [global]: Disable

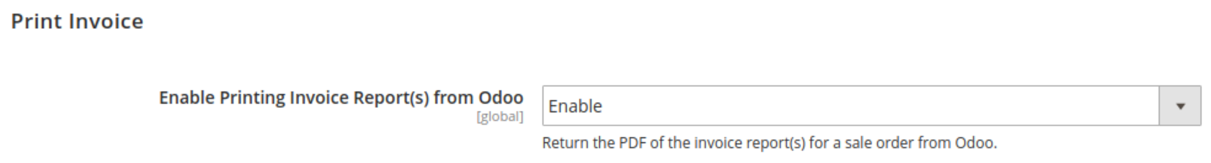
Enable or disable order payment status import.

Order status [global]: Pending

The selected status will be used to filter orders for payment status update.

e. PRINT INVOICE

- **Enable Printing Invoice Report(s) from Odoo**
    - Enable or disable the return of the invoice report(s) for a sale order from Odoo

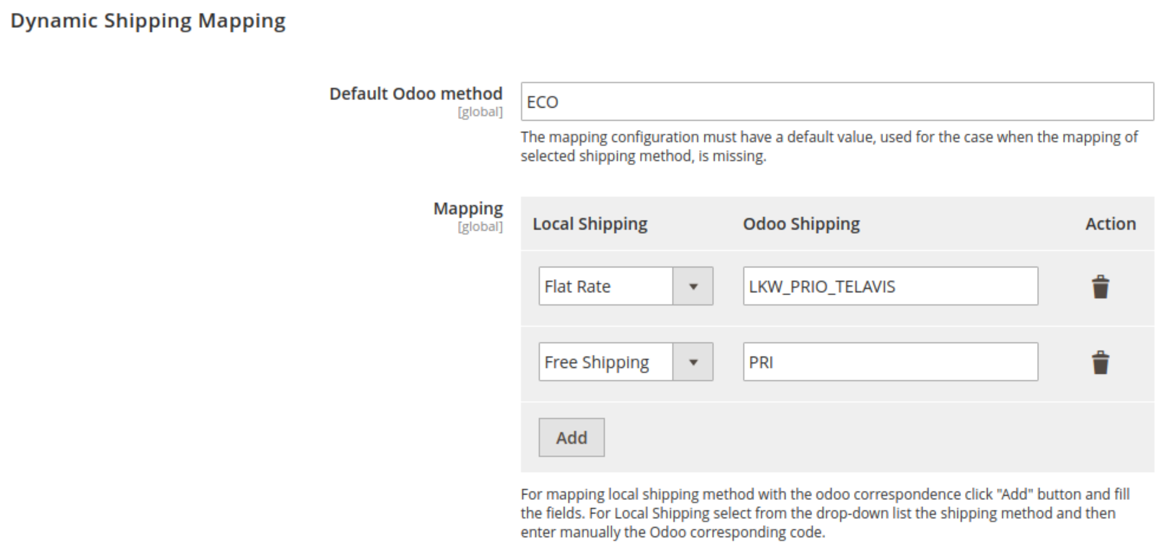

Print Invoice

Enable Printing Invoice Report(s) from Odoo [global]: Enable

Return the PDF of the invoice report(s) for a sale order from Odoo.

f. DYNAMIC SHIPPING MAPPING

- **Default Odoo method**
    - The mapping configuration must have a default value, used for the case when the mapping of selected shipping method, is missing.
- **Mapping**
    - Will map the local shipping method with Odoo correspondence
    - For mapping local shipping method with the Odoo correspondence, click "Add" button and fill the fields.
    - For Local Shipping select from the drop-down list the shipping method and then enter manually the Odoo corresponding code.


Dynamic Shipping Mapping

Default Odoo method [global]: ECO

The mapping configuration must have a default value, used for the case when the mapping of selected shipping method, is missing.

Mapping [global]

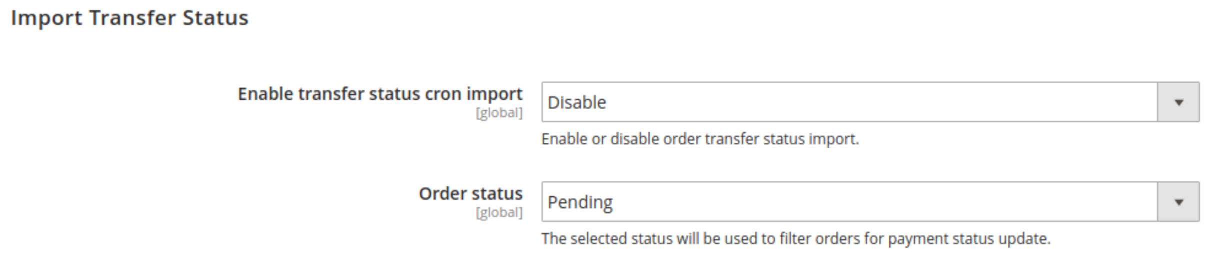
Local Shipping	Odoo Shipping	Action
Flat Rate	LKW_PRIO_TELAVIS	
Free Shipping	PRI	

For mapping local shipping method with the odoo correspondence click "Add" button and fill the fields. For Local Shipping select from the drop-down list the shipping method and then enter manually the Odoo corresponding code.

g. IMPORT TRANSFER STATUS

- **Enable transfer status cron import**
    - Enable or disable cron import of orders transfer status

- **Order status**
    - The selected status will be used to filter orders for transfer status update.

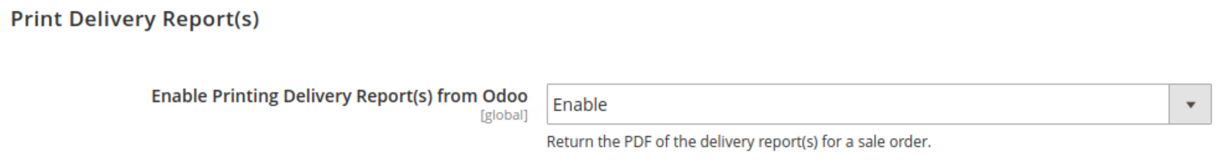

Import Transfer Status

Enable transfer status cron import [global]: Disable
Enable or disable order transfer status import.

Order status [global]: Pending
The selected status will be used to filter orders for payment status update.

h. PRINT DELIVERY REPORT(S)

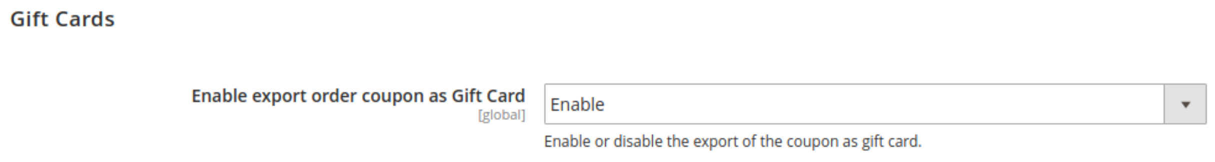
- **Enable Printing Delivery Report(s) from Odoo**
    - Enable or disable the return of the delivery report(s) for a sale order from Odoo


Print Delivery Report(s)

Enable Printing Delivery Report(s) from Odoo [global]: Enable
Return the PDF of the delivery report(s) for a sale order.

i. GIFT CARDS

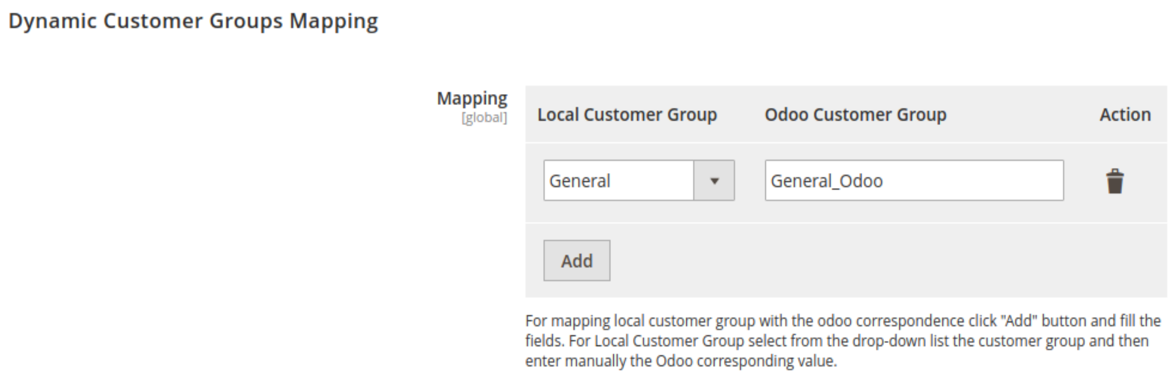
- **Enable export order coupon as Gift Card**
    - Enable or disable the export of the coupon as gift card
    - If is enabled, the system will check if the new order has a coupon attached or not. If the coupon is present, it will be sent to Odoo


Gift Cards

Enable export order coupon as Gift Card [global]: Enable
Enable or disable the export of the coupon as gift card.

j. DYNAMIC CUSTOMER GROUPS MAPPING

- **Mapping**
    - Will map the local customer groups with Odoo correspondence
    - For mapping local customer group with the odoo correspondence click "Add" button and fill the fields.
    - For Local Customer Group select from the drop-down list the customer group and then enter manually the Odoo corresponding value.


Dynamic Customer Groups Mapping

Mapping [global]

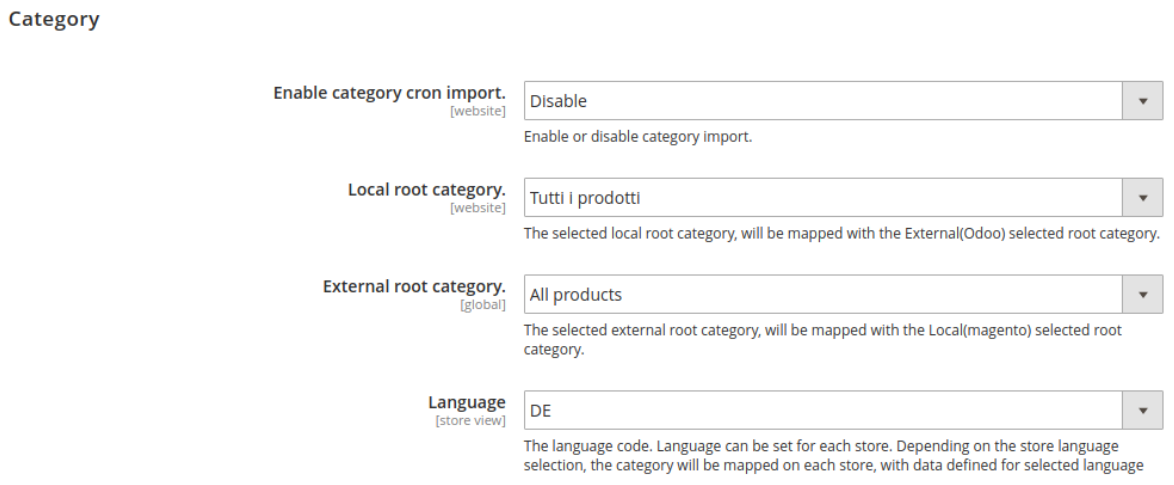
Local Customer Group	Odoo Customer Group	Action
General	General_Odoo	

For mapping local customer group with the odoo correspondence click "Add" button and fill the fields. For Local Customer Group select from the drop-down list the customer group and then enter manually the Odoo corresponding value.

3.3 SECTION > CATALOG

a. CATEGORY

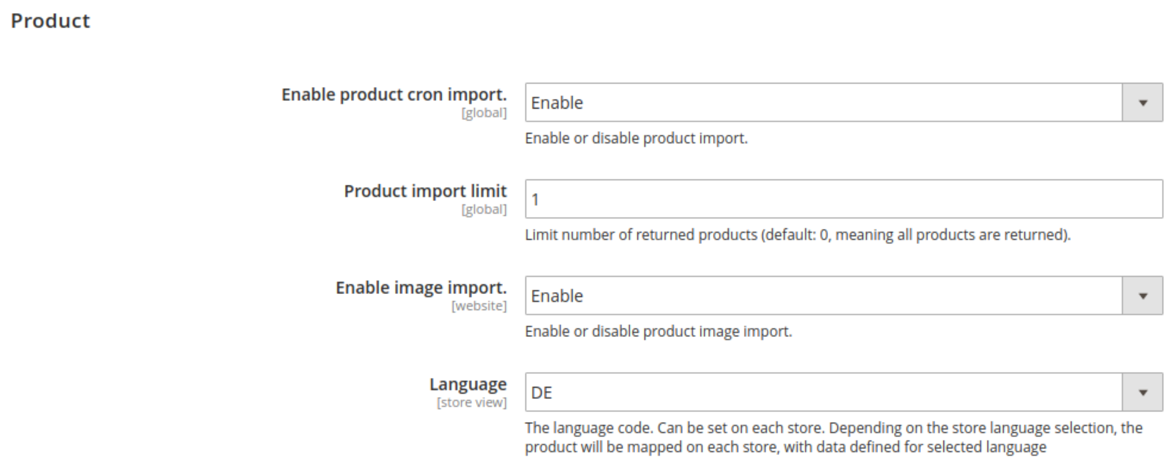
- **Enable category cron import**
    - Enable or disable the category import, using cron jobs
- **Local root category**
    - Select the local root category
    - The selected local root category, will be mapped with the External (Odoo) selected root category
- **External root category**
    - Select the Odoo root category
    - The selected external root category, will be mapped with the Local (Magento) selected root category
- **Language**
    - Language can be set for each store
    - Depending on the store language selection, the category will be mapped on each store, with data defined for selected language


Category

Enable category cron import. [website]	Disable Enable or disable category import.
Local root category. [website]	Tutti i prodotti The selected local root category, will be mapped with the External(Odoo) selected root category.
External root category. [global]	All products The selected external root category, will be mapped with the Local(magento) selected root category.
Language [store view]	DE The language code. Language can be set for each store. Depending on the store language selection, the category will be mapped on each store, with data defined for selected language

b. PRODUCT

- **Enable product cron import**
  - Enable or disable product import using cron job
- **Product import limit**
  - Limit number of returned products (default: 0, meaning all products are returned).
- **Enable image import**
  - Enable or disable product image import
  - If the import is enabled, when a product is imported, the product images will be imported as well
- **Language**
  - Can be set on each store
  - Depending on the store language selection, the product will be mapped on each store, with data defined for selected language


Product

Enable product cron import. [global] Enable
Enable or disable product import.

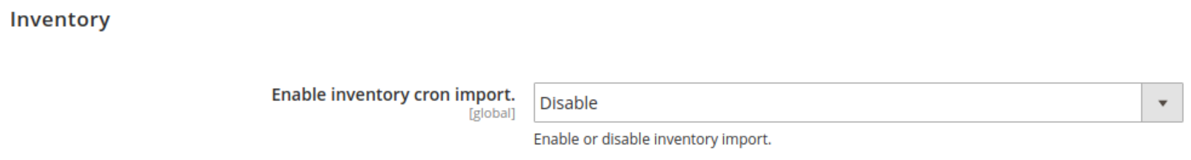
Product import limit [global] 1
Limit number of returned products (default: 0, meaning all products are returned).

Enable image import. [website] Enable
Enable or disable product image import.

Language [store view] DE
The language code. Can be set on each store. Depending on the store language selection, the product will be mapped on each store, with data defined for selected language

c. INVENTORY

- **Enable inventory cron import**
  - Enable or disable updating the product inventory using cron


Inventory

Enable inventory cron import. [global] Disable
Enable or disable inventory import.

4. CATEGORY IMPORT

All the categories are imported from Odoo ERP through cron job. In order to have a proper import, first thing to do, is to define in configuration (check 3.3.a.) the root categories on both sides (Odoo and Magento).
Each imported category is set on each store according to the selected language in configuration (check 3.3.a.). The language dependent fields are the name and the category description. There are available 3 translations: DE, FR, IT.

5. PRODUCT IMPORT

All the products are imported from Odoo ERP using cron job or manually, by using **Odoo Import** option **(**button) from product detail page. Manual import is available only after the product has been imported, and the product is available in the system. In order to trigger the products import using cron, **Enable product cron import** must be enabled in configuration (check 3.3.b.). Each product can be set to have different languages for each store. There are available 3 translations: DE, FR, IT (check 3.3.b.).

5.1. IMAGE IMPORT

Each product can have one or more images attached. To trigger the product image import, the option must be enabled in configuration (check 3.3.b.).
Before setting the new product images, the system checks if the product has attached any image(s) and if so, it will detach them from product and remove them from the disk. After the product images have been removed, the new images will be set on the product.

5.2. INVENTORY (STOCK)

When product is imported, the inventory is set as well, but in case we need to synchronize the product stocks from time to time, the cron job can be enabled in configuration (check 3.3.b.).

5.3. PRODUCT TAX CLASS

Each Magento product must have set a tax class. If the product tax class is not configured (check 3.2.a.), when import take place, the product will have the tax class with id 0(None).
If the product tax class is mapped, the product tax class will be set depending on the Odoo tax class correspondence.

5.4. ODOO CONNECT

A new tab has been added to the product detail page called “Odoo Connect”. The tab contains the following fields:

- Ean13
- Width
- Height
- Diameter
- Length
- Weight Net
- UOM Name
- Volume
- Manufacturer Website
- Sale Delay
- Odoo id

6. ORDER EXPORT

A registered customer order is exported to Odoo ERP in 3 ways:

- Right after an order is created
- Using cron job
- Manually export by clicking on the **Odoo Export** button from order details page

An order from guest customer can be exported in 2 ways:

- Using cron job
- Manually export by clicking on the **Odoo Export** button from order details page

6.1. ORDER EXPORT USING CRON

For exporting the order using the cron job, the module must be configured (check 3.2.a.).
First the system checks if the order cron is enabled. If is enabled, the system will export only those orders which have the status defined in the **Export order status** text area in configuration (check 3.2.a.).
Depending on the order export result, successful or failed, the new order status will be set to be the one defined in configuration(check 3.2.a.) for each type of the result(successful or failed).

6.2. PAYMENTS MAPPING

When exporting an order, the system must check which Odoo payment method is mapped to match the selected order method. If a mapped method is found, the module will set the payment method to be the mapped one, in the data sent to Odoo ERP. If the mapping does not include the payment method of the selected order, the module will return the **Default Odoo method** from configuration (check 3.2.c.)

6.3. SHIPPING MAPPING

The active local shipping methods must have a correspondence to Odoo shipping methods. The methods can be mapped by accessing the **Sales** section of **Epoint Swiss post Api** tab (check 3.2.f.)
When an order is exported, the data sent to the ERP, will contain the Odoo method mapped with selected order method. In case there is no mapping for the order shipping method, the Default Odoo method will be sent.

6.4. CUSTOMER GROUPS MAPPING

When an order/account is exported to Odoo ERP, the data must contain the account_categories field. The field value will be the Odoo correspondence of the Magento customer group.
If there is no mapping for the customer group, the value will be set to be the Magento customer group code (Ex. General, Retailer...etc).

6.5. SALES PRODUCT ORDER LINE

This mapping is used when importing a product or exporting an order.
When an order is exported, the module verify if any Odoo tax class has been mapped with Magento product tax class. If has been setup, then it will return the mapped value. If not it will return the **Default Odoo tax class**.

7. API LOGGING

Logging Api communication can be done using the 2 system implemented
a. Using a log file
b. Using emails

The email logging system has the **Debug** and **Error** level. The Debug emails will receive the successful and failed requests while the Error emails will receive only the failed requests.
For configuration setup please check 3.1.b.

8. CRON

The extension itself contains the needed configurations of the cron jobs that need to run periodically. The only thing which is needed, is to have the cron function enabled on the server.

Optional:
In order to view the settings and to manually run the cron via the admin interface, Magento Community Edition online shops can install the community plugin Cron Scheduler which is available for Magento CE 2+.

9. CHECK CUSTOMER CREDIT

The module implements a console command for checking the customer credit. To use it navigate to {Magento root} and use the following command:

```
./bin/magento epoint-swisspostapi:checkCustomerCredit <customer_id> <credit_amount>
```

<customer_id> - the magento ID of the customer
<credit_amount> - credit value that needs to be verified