

# Odoo – Drop Shipping API

## 1. Setting up the default company for generic sale order creation

### 1.1. Setting up default company for generic sale order creation

Company > Default company for generic sale order creation

Companies / My Company (San Francisco)

EDIT CREATE Print Action 2 / 2 < >

Company Name  
**My Company (San Francisco)**  
Your logo

General Information

Contact

My Company (San Francisco)

Address

250 Executive Park Blvd, Suite 3400  
San Francisco California (US) 94134  
United States

Tax ID

Company Registry

Currency

USD

Default company for generic sale order creation

Phone

+1 (650) 555-0111

Mobile

Email

info@yourcompany.com

Website

http://www.example.com

Parent Company

Company Favicon

## 2. Create sale order through API:

### 2.1. Creating Consumer Key and Consumer Secret

Settings > Users > Auth2 > Create > Create with any Name and Description > Regenerate Consumer key and secret.

Users / Mitchell Admin

EDIT CREATE Action 2 / 2 < >

SEND PASSWORD RESET INSTRUCTIONS NEVER CONNECTED CONFIRMED

Name  
**Mitchell Admin**

Email Address  
**admin**

Related Partner  
YourCompany, Mitchell Admin

Access Rights

Preferences

Account Security


User Type  
Internal User

19 Groups

Auth2

400 Access Rights

58 Record Rules



Users / Mitchell Admin / Auth2 / New

SAVE DISCARD

NEW

User

Mitchell Admin

Description

Descriptions...

Endpoints

/oauth2/auth  
/oauth2/access\_token

Key and Secret

OAuth2 Access Token

Consumer Key(API)

Consumer Secret(API)

REGENERATE CONSUMER KEY AND SECRET

## 2.2. Generating the Access token

HTTP HEADERS	<code>{"Accept": "*/*"}</code>
ENDPOINT	<code>/oauth2/access_token</code>
METHOD	POST
REQUEST BODY	<code>{'client_id': {Consumer Key}, 'client_secret': {Consumer Secret}, 'grant_type': 'client_credentials'}</code>
RESPONSE SUCCESS	<code>{"access_token": "{access_token}", "token_type": "bearer", "access_token_validity": "{token_valid_until}", "refresh_token": false}</code>
RESPONSE ERROR	<code>{"error": "Unauthorized_client", "error_description": "Invalid Consumer ID"}</code>

### Request parameter description

PARAMETER	TYPE	DESCRIPTION	MANDATORY/OPTIONAL
client_id	CHAR	String value that uniquely identifies the client application in the authorization server	MANDATORY
client_secret	CHAR	This is a secret value that can be used with client id to authenticate with the authorization server	MANDATORY
grant_type	CHAR	This tells the token endpoint that the application is using the Authorization Code grant type.	MANDATORY

## 2.3. Process sale order:

ENDPOINT	<code>/api/process_sale_order</code>
METHOD	POST
HTTP HEADERS	<code>{"Content-Type": "application/json", "Authorization": "Bearer {access_token}"}</code>
REQUEST BODY	<pre>{   "customer_details": {     "code": {customer_partner_code},     "name": {customer_name},     "address_line_1": {customer_address_line_1},     "address_line_2": {customer_address_line_2},     "city": {customer_city},     "state": {customer_state},     "zip": {customer_zip},     "country": {customer_country},     "email": {customer_email},     "phone": {customer_phone},     "mobile": {customer_mobile},     "warehouse_code": {warehouse_code},   },   "order_details": [{     "product_details": {       "code": {product_code},       "name": {product_name},       "product_type": {product_type},       "sale_price": {product_price},       "product_category": {product_category}     }   ] }</pre>

	<pre>         "currency": {currency},         "quantity": {product_quantity},         "unit_price": {unit_price},         "taxes": {product_taxes}       },     ],   } </pre>
RESPONSE SUCCESS	<pre>{'success': True, 'message': 'Sale order XXXX created successfully', 'status': 200}</pre>
RESPONSE ERROR	<pre>{'error': '{error text}', 'status': 400}</pre>

### Request parameter description

PARAMETER	TYPE	DESCRIPTION	MANDATORY/ OPTIONAL
customer_partner_code	CHAR	Customer's unique partner code	MANDATORY
customer_name	CHAR	Customer's name	OPTIONAL
customer_address_line_1	CHAR	Customer's address line 1	MANDATORY
customer_address_line_2	CHAR	Customer's address line2	OPTIONAL
customer_city	CHAR	Customer's city	OPTIONAL
customer_state	CHAR	Customer's state	OPTIONAL
customer_zip	CHAR	Customer's zipcode	OPTIONAL
customer_country	CHAR	Customer's country	OPTIONAL
customer_email	CHAR	Customer's email	MANDATORY
customer_phone	CHAR	Customer's phone number	MANDATORY
customer_mobile	CHAR	Customer's mobile number	OPTIONAL
warehouse_code	CHAR	Warehouse code	MANDATORY
product_code	CHAR	Unique product code	MANDATORY
product_name	CHAR	Product name	OPTIONAL
product_type	CHAR	Product type ['consu' for consumable, 'service' for service, 'product' for storable]	OPTIONAL
product_price	FLOAT	Sale price of the product	OPTIONAL
product_category	CHAR	Category of the product	OPTIONAL
currency	Char	Currency for the product	OPTIONAL
quantity	FLOAT	Ordered product quantity	MANDATORY
unit_price	FLOAT	Unit price of the product	MANDATORY
taxes	FLOAT	Product taxes	OPTIONAL

## Create sale order with XML file.

Download sample xml file.

Sale > Configuration > Settings > Download Sample xml

Settings Search...

**SAVE** **DISCARD**

**General Settings**

**Sales**

Magento

Magento Websites

Magento Storeview

Inventory

Invoicing

**Invoicing**

**Invoicing Policy** ⓘ  
Quantities to invoice from sales orders

☒ Invoice what is ordered  
☐ Invoice what is delivered

**Automatic Invoice**  
☐ Generate the invoice automatically when the online payment is confirmed

**Down Payments** ⓘ  
Product used for down payments

**Connectors**

☐ **Amazon Sync** ⓘ  
Import Amazon orders and sync deliveries

**E-Commerce API**

**Folder for data files**  
Folder that contains .xml or .csv file with the sale order data  
C:\Users\pushpitha\OneDrive - elitt

[Sample .xml](#) [Sample .csv](#)

XML file format (You can add multiple sale orders lines using <orderline/> tag)

```
<?xml version="1.0"?>
<data>
  <orderReference>{order_reference[OPTIONAL]}</orderReference>
  <customerDetails>
    <code>{customer_code[MANDATORY]}</code>
    <name>{customer_name[OPTIONAL]}</name>
    <addressLine1>{customer_address_line_1[MANDATORY]}</addressLine1>
    <addressLine2>{customer_address_line_2[OPTIONAL]}</addressLine2>
    <city>{customer_city[OPTIONAL]}</city>
    <state>{customer_state[OPTIONAL]}</state>
    <zip>{customer_zip[OPTIONAL]}</zip>
    <country>{customer_country[OPTIONAL]}</country>
    <email>{customer_email[MANDATORY]}</email>
    <phone>{customer_phone_number[MANDATORY]}</phone>
    <mobile>{customer_mobile_number[OPTIONAL]}</mobile>
    <warehouse_code>{warehouse_code[MANDATORY]}</warehouse_code>
  </customerDetails>
  <orderDetails>
    <orderLine>
      <productDetails>
        <code>{unique_product_code[MANDATORY]}</code>
        <name>{product_name[OPTIONAL]}</name>
        <productType>{product_type[OPTIONAL]}</productType>
        <salePrice>{product_price[OPTIONAL]}</salePrice>
        <productCategory>{product_category[OPTIONAL]}</productCategory>
      </productDetails>
      <currency>{product_currency[OPTIONAL]}</currency>
      <quantity>{product_quantity[MANDATORY]}</quantity>
      <unitPrice>{product_unit_price[MANDATORY]}</unitPrice>
      <taxes>{product_taxes[OPTIONAL]}</taxes>
    </orderLine>
  </orderDetails>
</data>
```

**Add the folder path on Odoo server.**

Sales > Configuration > Folder for XML file > Add the folder path that contains the xml files.

The screenshot shows the Odoo Settings interface. On the left is a dark sidebar with a menu containing 'General Settings', 'Sales', 'Magento', 'Magento Websites', 'Magento Storeview', 'Inventory', and 'Invoicing'. The 'Invoicing' option is selected. The main content area is titled 'Invoicing' and contains three sections: 'Invoicing Policy', 'Connectors', and 'E-Commerce API'. The 'Invoicing Policy' section has two radio buttons: 'Invoice what is ordered' (selected) and 'Invoice what is delivered'. There is also an 'Automatic Invoice' checkbox which is unchecked. The 'Connectors' section has an 'Amazon Sync' checkbox which is unchecked. The 'E-Commerce API' section has a 'Folder for data files' field with a description 'Folder that contains .xml or .csv file with the sale order data' and a text input field containing '/opt/odoo/data\_folder/'. Below this field are two download links: 'Sample .xml' and 'Sample .csv'.

**3.4.** Put the generated XML file that contains sale order data into the folder that set up in the Odoo server. **NOTE: Need to add separate XML files for each sale order**

**3.5.** Odoo server will check the folder and create sale orders for each XML files with the Scheduled action that is configured in Odoo back-end.

Action Name

**E-Commerce API: Create sale orders from xml file**

Model Sales Order

Scheduler User OdooBot

Execute Every 1Days

Active 

Next Execution Date 03/02/2023 15:58:38

Number of Calls -1

Priority 5

Repeat Missed ☒

Python Code Security Help

model.cron\_create\_sale\_orders()

## 4. Create sale orders with .csv files.

### 4.1. Download sample csv file.

Sale &gt; Configuration &gt; Settings &gt; Download Sample csv

Settings

Search...

SAVE

DISCARD

Unsaved changes

- General Settings
- Sales
- Magento
- Magento Websites
- Magento Storeview
- Inventory
- Invoicing

#### Invoicing

##### Invoicing Policy ?

Quantities to invoice from sales orders

- ☒ Invoice what is ordered
- ☐ Invoice what is delivered

☐

##### Automatic Invoice

Generate the invoice automatically when the online payment is confirmed

##### Down Payments ?

Product used for down payments

#### Connectors

##### ☐ Amazon Sync ?

Import Amazon orders and sync deliveries

#### E-Commerce API

##### Folder for data files

Folder that contains .xml or .csv file with the sale order data

[Sample .xml](#) [Sample .csv](#)

### 4.2. CSV file format

	A	B	C	D	E	F	G	H	I	J	K	L
1	order_reference	customer_details/code	customer_details/address_line_1	customer_details/email	customer_details/phone	order_details/product_details/code	order_details/quantity	order_details/unit_price				
2	Order Reference	Customer code	Customer address line 1	Customer Email	Customer Phone	Product Code	Order Quantity	Order unit price				
3						Product Code	Order Quantity	Order unit price				
4	Order Reference	Customer Code	Customer Address line 1	Customer email	Customer Phone	Product Code	Order Quantity	Order unit price				
5						Product Code	Order Quantity	Order unit price				
6												
7												
8												
9												
10												
11												
12												
13												

4.3. Refer point 3.4 and 3.5 for import folder and schedule action. You can import multiple sale order data with csv import option. Please check the sample file for multiple sale order format (6.3 or 4.2).

### 4.4. Parameter Description

Name	Description	Mandatory/ Optional
order_reference	Unique reference for the order	Optional
customer_details/code	Customer's unique partner code	Mandatory

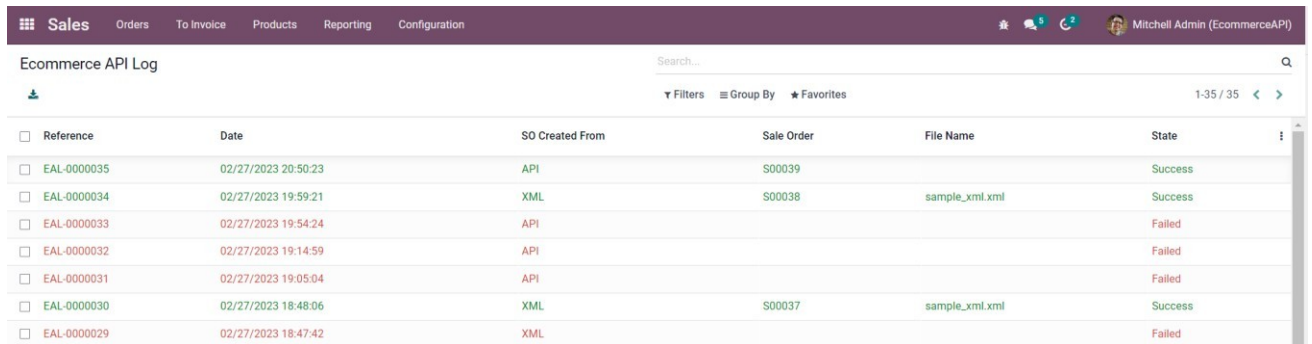
customer_details/address_line_1	Customer's address	Mandatory
customer_details/email	Customer's email	Mandatory
customer_details/phone	Customer's phone	Mandatory
customer_details/warehouse_code	Warehouse code	Mandatory
order_details/product_details/code	Order product code	Mandatory
order_details/quantity	Order product quantity	Mandatory
order_details/unit_price	Order product unit price	Mandatory

## 5. Logs and trace-back:

5.1. There is a log line created in Odoo back-end for each API calls and each XML files that are executing from scheduled action.

### 5.1.1. How to check the logs?

Enable Developer mode > Sales > Configuration > API-Log



Reference	Date	SO Created From	Sale Order	File Name	State
EAL-000035	02/27/2023 20:50:23	API	S00039		Success
EAL-000034	02/27/2023 19:59:21	XML	S00038	sample_xml.xml	Success
EAL-000033	02/27/2023 19:54:24	API			Failed
EAL-000032	02/27/2023 19:14:59	API			Failed
EAL-000031	02/27/2023 19:05:04	API			Failed
EAL-000030	02/27/2023 18:48:06	XML	S00037	sample_xml.xml	Success
EAL-000029	02/27/2023 18:47:42	XML			Failed

Green – Success and Red - Failed.

### 5.1.2. Field Description

Field	Description
Reference	Unique sequence for identify the log line
Date	Executed date
So Created From	Which type of data the sale order is created from, either API or XML.
Sale Order	Sale order related to the log line if the try is successful
File Name	Related xml file name
State	Try is success or failed
Status Message	Success or failed response message
JSON Body	JSON object with the data



**EAL-0000034**

Date	02/27/2023 19:59:21	JSON body	{'company_details': {'code': '123456', 'name': 'Test Company'}, 'customer_details': {'code': '234567', 'name': 'Test Customer', 'address_line_1': 'Test address 1', 'address_line_2': 'test address 2', 'city': 'Test City', 'state': 'Test State', 'zip': '21490', 'country': 'United States', 'email': 'test@gmail.com', 'phone': '2093784390392', 'mobile': '2928398239933'}, 'order_details': [{'product_details': {'code': 'FURN_6666', 'name': 'TV', 'product_type': 'consu', 'sale_price': '50000', 'product_category': 'All'}, 'currency': 'USD', 'quantity': '2', 'unit_price': '50000', 'taxes': None}, {'product_details': {'code': 'E-COM11', 'name': 'Drawer', 'product_type': 'consu', 'sale_price': '50500', 'product_category': 'All'}, 'currency': 'USD', 'quantity': '3', 'unit_price': '50500', 'taxes': None}]}
Status Message	{'success': True, 'message': 'Sale order S00038 created successfully', 'status': 200}		
Sale Order	S00038		
SO Created From	XML		
File Name	sample_xml.xml		

## 6. Samples

### 6.1. Python

## 7. `import requests` `import json`

```
# =====Get access token=====
payload = {
    'client_id': 'bjqk6ydeZmxM30upYGWH7EqmbEZLGQ',
    'client_secret': 'PwVB9Zn7em8akn63yoXvGdSFjAWyk6',
    'grant_type': 'client_credentials',
}

response = requests.post(
    '{BASE_URL}/oauth2/access_token',
    data=payload,
    headers={'Accept': '*/*'})
token_data = response.json()

# =====Create sale order=====
so_payload = {
    "company_details": {
        "code": 123456,
        "name": "Test Company"
    },
    "customer_details": {
        "code": 234567,
        "name": "Test Customer",
        "address_line_1": "Test address 1",
        "address_line_2": "test address 2",
        "city": "Test City",
        "state": "Test State",
        "zip": 21490,
        "country": "United States",
        "email": "test@gmail.com",
        "phone": 2093784390392,
        "mobile": 2928398239933,
        "warehouse_code": DROP1MALAG,
    },
    "order_details": [{
        "product_details": {
            "code": "FURN_6666",
            "name": "TV",
            "product_type": "consu",
            "sale_price": 50000,
            "product_category": "All"
```

}, {

```
},
"currency": {
  "currency": "USD",
  "quantity": 2,
  "unit_price": 50000,
  "taxes": False
  "product_details": {
    "code": "E-COM11",
    "name": "Drawer",
    "product_type": "consu",
    "sale_price": 50500,
    "product_category": "All"
  },
  "currency": "USD",
  "quantity": 3,
  "unit_price": 50500,
  "taxes": False
}
}]
```

```

response = requests.post(
    '{BASE_URL}/api/process_sale_order',
    data=json.dumps(so_payload),
    headers={
        'Content-Type': 'application/json',
        'Authorization': f"Bearer {token_data['access_token']}"
    }
)
print(response.json())

```

## 6.2.XML

```

<?xml version="1.0"?>
<data>
  <orderReference>12345678</orderReference>
  <customerDetails>
    <code>234567</code>
    <name>Test Customer</name>
    <addressLine1>Test address 1</addressLine1>
    <addressLine2>test address 2</addressLine2>
    <city>Test City</city>
    <state>Test State</state>
    <zip>21490</zip>
    <country>United States</country>
    <email>test@gmail.com</email>
    <phone>2093784390392</phone>
    <mobile>2928398239933</mobile>
    <warehouse_code>DROP1MALAG</warehouse_code>
  </customerDetails>
  <orderDetails>
    <orderLine>
      <productDetails>
        <code>FURN_6666</code>
        <name>TV</name>
        <productType>consu</productType>
        <salePrice>50000</salePrice>
        <productCategory>All</productCategory>
      </productDetails>
      <currency>USD</currency>
      <quantity>2</quantity>
      <unitPrice>50000</unitPrice>
      <taxes></taxes>
    </orderLine>
    <orderLine>
      <productDetails>
        <code>E-COM11</code>
        <name>Drawer</name>
        <productType>consu</productType>
        <salePrice>50500</salePrice>
        <productCategory>All</productCategory>
      </productDetails>
      <currency>USD</currency>
      <quantity>3</quantity>
      <unitPrice>50500</unitPrice>
      <taxes></taxes>
    </orderLine>
  </orderDetails>

```

