



User Manual

PostFinance POS Payment Terminal Plugin for Odoo 19.0

Table of Contents

1	Introduction.....	4
2	Features.....	5
3	Advantages.....	6
4	Prerequisites	7
4.1	Odoo Requirements	7
4.2	PostFinance Requirements	7
4.3	Required REST API Credentials.....	8
5	Installation.....	8
5.1	On-Premise Installation	8
5.2	Odoo.sh Installation.....	10
6	Configuration	13
6.1	Configure the PostFinance Payment Provider	13
6.2	Configure the POS Payment Method.....	14
6.3	Add the Payment Method to the POS.....	15
7	Processing a Payment.....	16
7.1	Create a POS Order	16
7.2	Process Payment Through the Terminal.....	17
7.3	Complete the POS Order	18
7.4	Customer Payment Email.....	19
8	Payment Failure and Cancellation	20
9	Refunds.....	21
9.1	Refund From the POS Order.....	21
9.2	Refund From the POS Store.....	23
9.3	Refund Requirements	25
9.4	Refund Validation Messages.....	26
10	Transaction Information	28
11	Troubleshooting	30



11.1	"External dependency not satisfied" / Module Not Installable	30
11.2	Payment Provider Cannot Be Saved	30
11.3	Plugin Does Not Appear in Apps	31
11.4	PostFinance POS Option Does Not Appear	31
12	Technical Requirements and Compatibility.....	31
13	Change Log / Release Notes	31
14	Support.....	32

1 Introduction

The **PostFinance POS Payment Terminal Plugin** is officially developed and maintained by **PIT Solutions**, the technology partner behind the integration for **PostFinance**. It integrates PostFinance Checkout payment terminals with Odoo Point of Sale.

The plugin transfers payment requests from Odoo POS to the configured PostFinance payment terminal, allowing customers to complete payments directly on the terminal.

This integration eliminates the need for the cashier to manually enter the payment amount on the payment terminal and provides payment status information back to Odoo POS.

The PostFinance POS Payment Terminal Plugin addresses these issues by:

- **Official Development:** Developed and maintained by PIT Solutions for PostFinance.
- **Automating Payment Processing:** Eliminating manual data entry and reducing the risk of human errors.
- **Streamlining the Checkout Process:** Providing a faster and more efficient payment experience for both merchants and customers.
- **Enhancing Accuracy:** Ensuring that payment amounts are accurately transferred from the Odoo POS to the PostFinance payment terminal.
- **Increasing Efficiency:** Optimizing the checkout process and reducing operational costs.
- **Enhancing Customer Satisfaction:** Delivering a faster and more convenient payment experience.
- **Ensuring Compliance:** Benefiting from PostFinance's adherence to industry security standards.

2 Features

The plugin provides the following features:

- **POS Terminal Payments**

Payment amounts are transferred from Odoo POS to the configured PostFinance payment terminal. This eliminates the need for the cashier to manually enter the payment amount on the terminal.

- **Payment Status Handling**

The Odoo POS interface receives the payment status from the PostFinance payment process and displays the result to the cashier.

- **Payment Cancellation**

The plugin supports handling cancelled or failed payment transactions.

- **Refunds**

Refunds are supported for POS orders paid through PostFinance.

- **Payment Transaction Tracking**

Payment transactions can be tracked in both:

- Odoo
- PostFinance Back Office

- **Customer Payment Notifications**

When a customer is selected for the POS order, the plugin can send an email notification containing payment information.

- **Supported Payment Connectors**

The plugin supports the following payment connectors:

- American Express
- Dankort
- Diners Club
- Mastercard

- Visa

3 Advantages

The PostFinance POS Payment Terminal Plugin provides the following advantages for Odoo POS merchants:

- **Reduced Checkout Time**
 - **Faster Payments:** Automates the transfer of the payment amount from Odoo POS to the PostFinance payment terminal, reducing manual steps during checkout.
 - **Improved Customer Experience:** Helps provide a faster and smoother checkout experience.
- **Easy Integration**
 - **Seamless Odoo POS Integration:** Integrates the PostFinance payment terminal into the Odoo POS payment workflow.
 - **Simple Configuration:** Provides the required configuration options to set up the payment integration.
- **Improved Payment Accuracy**
 - **Reduced Manual Errors:** Eliminates the need to manually enter payment amounts into the terminal, reducing the risk of incorrect amounts.
 - **Accurate Transaction Processing:** Ensures that the payment amount initiated from Odoo POS is passed to the payment terminal.
- **Simplified Reconciliation**
 - **Accurate Payment Records:** Payment transaction results are recorded in Odoo POS.
 - **Easier Reconciliation:** Reduces manual effort when matching POS orders with payment transactions.
- **Secure Payment Processing**
 - **Secure Transactions:** Payment processing is handled through the PostFinance payment infrastructure.

- **Industry-Standard Payment Security:** Benefits from the security mechanisms and payment standards supported by PostFinance.
- **Refund Support**
 - **POS Order Refunds:** Supports refund processing for POS orders through the integrated payment workflow.

4 Prerequisites

Before installing the plugin, ensure that the following requirements are available.

4.1 Odoo Requirements

The following Odoo applications are required:

- Point of Sale
- Payment
- Invoicing

The plugin supports **Odoo 19.0 Community and Enterprise Edition**.

4.2 PostFinance Requirements

You must have:

- An active PostFinance Checkout account.
- A PostFinance payment terminal ready for use.
- A PostFinance Checkout Space.
- REST API credentials.
- Access to the PostFinance Back Office.

4.3 Required REST API Credentials

The following credentials are required:

- **Terminal Identifier:** REST API identifier of the physical terminal
- **User ID:** REST API User ID
- **Space ID:** REST API Space ID
- **Application Key:** REST API Application Key

5 Installation

The installation procedure depends on whether Odoo is hosted **on-premise** or on **Odoo.sh**.

5.1 On-Premise Installation

Step 1 – Copy the Plugin to the Addons Directory

Copy the following module to an Odoo custom addons directory: *pits_pos_postfinance*

For example: */opt/odoo19/custom-addons/pits_pos_postfinance*

The custom addons directory must be included in the Odoo *addons_path*.

Important: Do not rename the *pits_pos_postfinance* directory.

Step 2 – Install Python Dependencies

The plugin contains a *requirements.txt* file:

pits_pos_postfinance/requirements.txt

The file contains the Python packages required by the plugin.

The current dependencies include:

- *postfinancecheckout*
- *cairosvg*
- *Pillow*
- *Requests*

Install the dependencies using the **same Python environment used to run Odoo**.

Linux / macOS

Activate the Odoo virtual environment:

```
source /path/to/odoo/venv/bin/activate
```

Install the dependencies:

```
pip install -r /path/to/addons/pits_pos_postfinance/requirements.txt
```

For example:

```
source /opt/odoo19/odoo-venv/bin/activate
```

```
pip install -r /opt/odoo19/custom-addons/pits_pos_postfinance/requirements.txt
```

Windows

Activate the Odoo virtual environment:

```
& C:\path\to\odoo\venv\Scripts\Activate.ps1
```

Then install the dependencies:

```
pip install -r C:\path\to\addons\pits_pos_postfinance\requirements.txt
```

Important: Do not install the packages into a different Python installation from the one used by Odoo.

Step 3 – Verify Python Dependencies

Run the following command using the same Python environment used by Odoo:

```
python -c "import postfinancecheckout, cairosvg, PIL, requests; print('OK')"
```

If the command returns:

```
OK
```

the required Python packages have been successfully installed.

If an import error is displayed, install the missing dependency before proceeding with the Odoo module installation.

Cairo System Dependency

The *cairosvg* package requires the Cairo system library.

On Debian/Ubuntu systems, if a Cairo-related installation error occurs, install the required system libraries:

```
sudo apt update
sudo apt install libcairo2 libpango-1.0-0 libgdk-pixbuf-2.0-0
```

On Windows, ensure that a compatible Cairo installation is available for the Python version used by Odoo.

5.2 Odoo.sh Installation

The installation procedure on Odoo.sh is different from an on-premise Odoo installation.

Important: Do not manually activate a Python virtual environment or run *pip install* on the Odoo.sh server. Odoo.sh installs the required Python packages automatically during the build process.

Step 1 – Add the Plugin to the Odoo.sh Git Repository

Add the *pits_pos_postfinance* module to the Git repository connected to your Odoo.sh project.

The plugin contains its own *requirements.txt* file: *pits_pos_postfinance/requirements.txt*

For the Odoo.sh deployment, the required Python packages must also be listed in the **root-level *requirements.txt* file of the Git repository.**

The repository should contain the following structure:

```
<Git repository root>/
|
├─ requirements.txt
|
└─ pits_pos_postfinance/
    ├── __init__.py
    ├── __manifest__.py
    ├── requirements.txt
    └─ ...
```

Important: The requirements.txt used for the Odoo.sh project dependency configuration is: *<Git repository root>/requirements.txt*

Do not rely only on: *pits_pos_postfinance/requirements.txt*

Step 2 – Configure the Root requirements.txt

Open the following file in the Git repository: *<Git repository root>/requirements.txt*

Add the following packages:

- *postfinancecheckout*
- *cairosvg*
- *Pillow*
- *requests*

Odoo.sh installs the packages listed in the project's *requirements.txt* during the build process.

Do not manually activate a Python virtual environment or run *pip install* on the Odoo.sh server.

Step 3 – Commit and Push the Changes

Commit both the PostFinance plugin and the root-level *requirements.txt* file to the Git repository.

For example:

```
git add requirements.txt pits_pos_postfinance/
```

```
git commit -m "Add PostFinance POS payment terminal plugin"
```

```
git push
```

Step 4 – Wait for the Odoo.sh Build

After the changes are pushed, Odoo.sh creates a new build.

During the build, Odoo.sh installs the Python packages listed in:

<Git repository root>/requirements.txt

Wait for the build to complete successfully before installing the Odoo module.

If the build fails, review the Odoo.sh build logs and Python package installation logs to identify the dependency causing the failure.

Step 5 – Install the Odoo Module

After the Odoo.sh build completes successfully:

1. Open the Odoo.sh environment.
2. Log in to Odoo with administrator privileges.
3. Enable **Developer Mode**.
4. Go to **Apps**.
5. Click **Update Apps List**.
6. Search for: **PostFinance POS Payment Terminal**
7. Click **Install**.

The technical module name is:

pits_pos_postfinance

Odoo.sh Dependency Structure

For clarity, the recommended repository structure is:

```
<Git repository root>/
|
|— requirements.txt
|
|— pits_pos_postfinance/
    |— __init__.py
    |— __manifest__.py
    |— requirements.txt
    |— models/
    |— views/
    |— static/
    |— ...
```

The root-level *requirements.txt* should contain:

- *postfinancecheckout*
- *cairosvg*
- *Pillow*
- *requests*

Odoo.sh will install these dependencies automatically as part of the build process.

6 Configuration

After installing the module, configure the PostFinance payment provider, POS payment method, and POS shop.

6.1 Configure the PostFinance Payment Provider

1. Go to: **Invoicing** → **Configuration** → **Payment Providers**

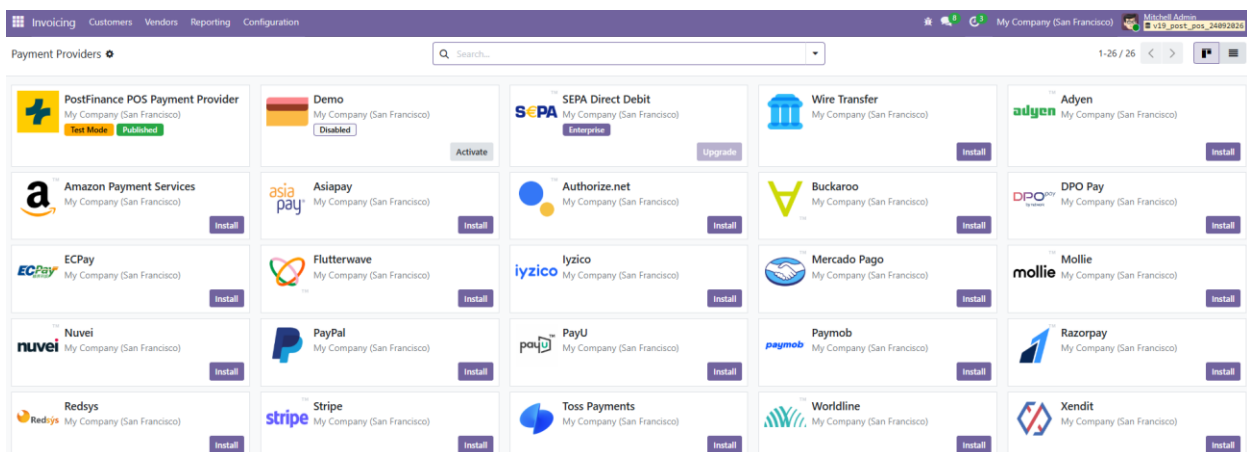


Figure 1: Payment Providers Listed in Odoo Backend

2. Open: **PostFinance POS Payment Provider**
3. Set the provider state to: **Test Mode**

While performing the initial configuration and testing.

Use: **Enabled**

Only when the configuration and terminal are ready for live transactions.

4. Enter PostFinance credentials

Enter the following information:

- **Terminal Identifier:** REST API identifier of the physical terminal
- **User ID:** REST API User ID
- **Space ID:** REST API Space ID
- **Application Key:** REST API Application Key
- **Send Status Email:** When this field is enabled, only then transaction-related emails, such as order confirmations, invoices, etc., will be sent through Odoo.

All four credentials are required when the provider is set to **Test Mode** or **Enabled**.

5. Click **Save**.

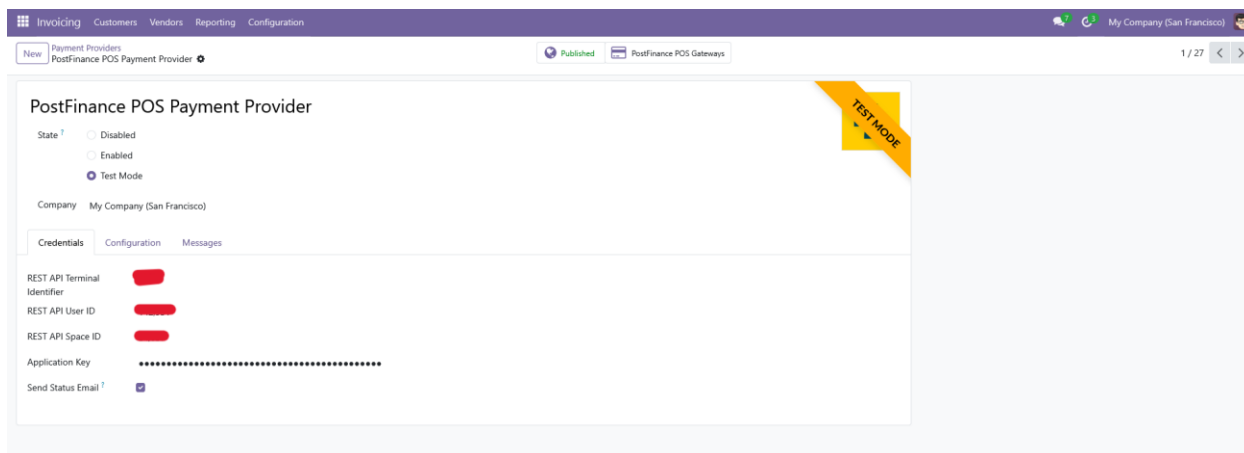


Figure 2: Odoo Backend Configuration for PostFinance Credentials

6.2 Configure the POS Payment Method

1. Go to: **Point of Sale** → **Configuration** → **Payment Methods**
2. Click **New**.

3. Configure the payment method as follows:

- Enter a name, for example: **PostFinance Terminal**
- Select the appropriate **Journal**.
- Enable **Use a Payment Terminal**.
- Select **PostFinance POS** as the payment terminal.
- Select the configured **PostFinance POS Payment Provider**.
- Save the payment method.

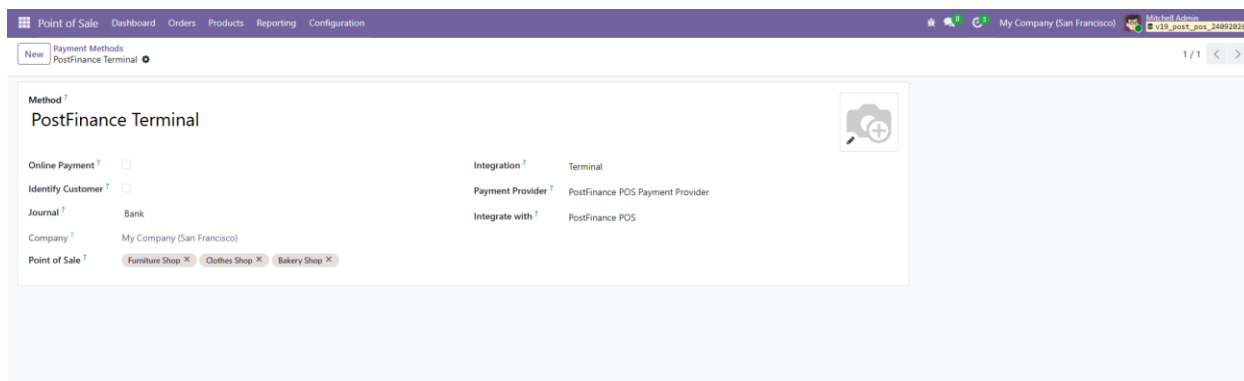


Figure 3: Create POS Payment Method for PostFinance

The **Payment Provider** field must be configured for payment methods using the PostFinance terminal.

6.3 Add the Payment Method to the POS

1. Go to: **Point of Sale** → **Configuration** → **Point of Sale**
2. Open the POS shop where the terminal will be used.
3. Under **Payment Methods**, add the PostFinance payment method.
4. Save the configuration.

If the POS session was already open before the configuration change, close the session and start a new session so that the updated payment terminal configuration is loaded.

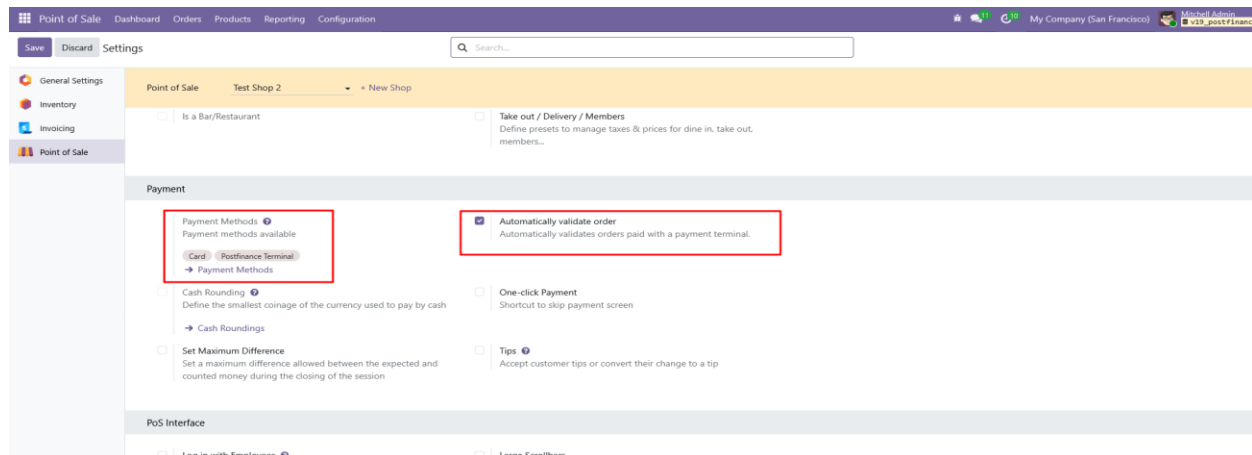


Figure 4: POS Payment Method Configuration for PostFinance

Note: Enable Automatically validate order if you want the POS order to be automatically validated after the payment is successfully completed through the payment terminal. If this option is disabled, the Validate button will be available on the payment screen, allowing the cashier to validate the order manually.

7 Processing a Payment

7.1 Create a POS Order

1. Open the POS application.
2. Select the required products.
3. Select or create the customer, if required.
4. Click Payment.

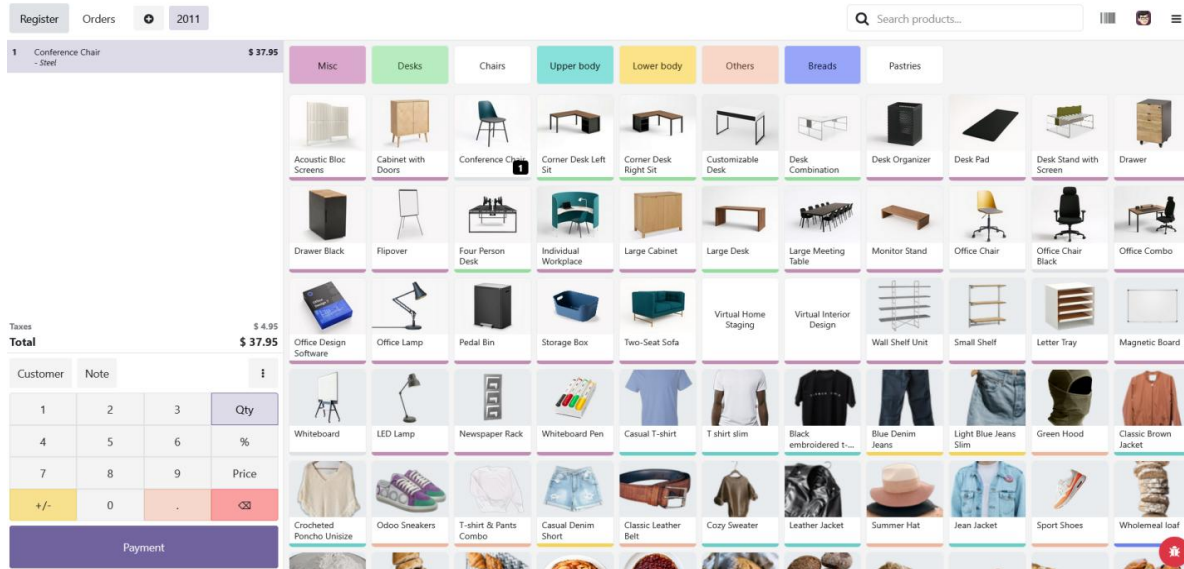


Figure 5: Odoo POS Product Screen

7.2 Process Payment Through the Terminal

1. Select the **PostFinance** payment method.
2. Initiate the payment.
3. Odoo sends the payment request to PostFinance.

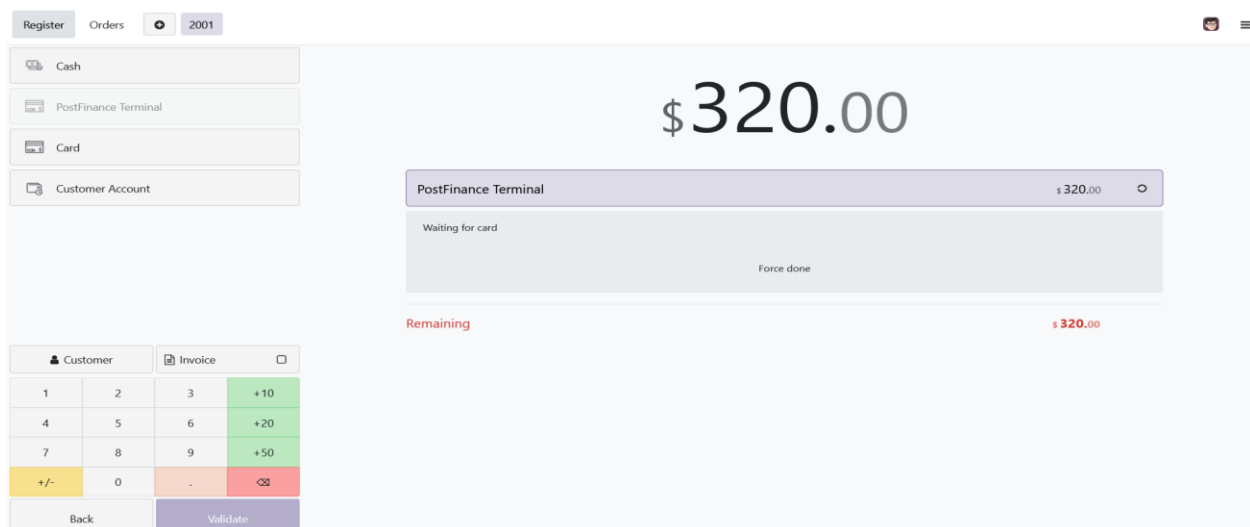


Figure 6: Odoo POS Payment Screen

4. The payment amount is displayed on the PostFinance terminal.
5. The customer selects the required card/payment option.
6. The customer enters the PIN when requested.
7. The payment is processed by the PostFinance terminal.

7.3 Complete the POS Order

After successful payment:

- Odoo displays **Payment Successful**.
- If **Automatically validate order** is enabled, the order is automatically validated.
- Otherwise, click **Validate** to complete the order.

After validation, the receipt can be sent or printed from the POS order screen.

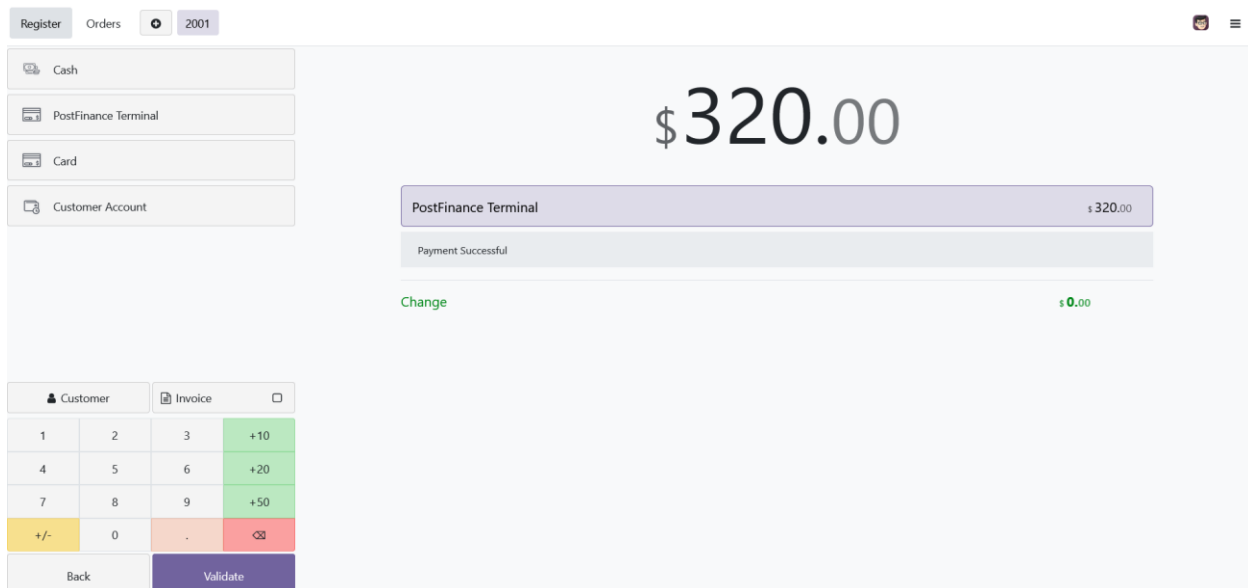


Figure 7: Odoo POS Payment Success Status

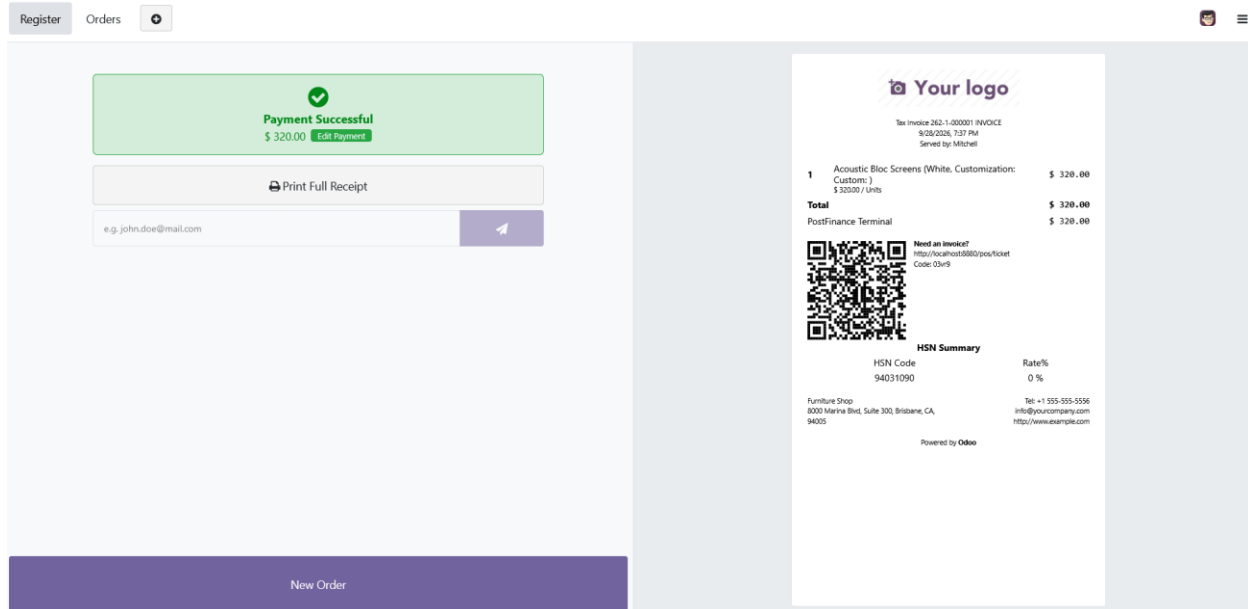


Figure 8: Odoo POS Order Validated Screen

7.4 Customer Payment Email

If a customer is associated with the POS order, the customer receives an email containing information about the successful payment.

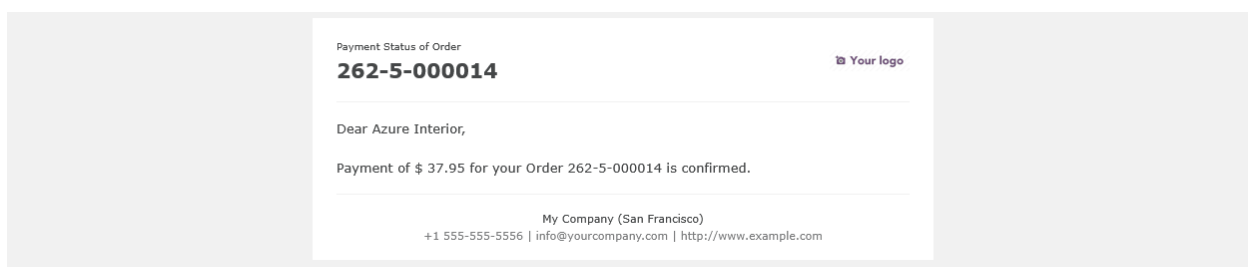


Figure 9: PostFinance Payment Successful Email to Customer

8 Payment Failure and Cancellation

If the payment fails, the payment status is displayed in Odoo and on the PostFinance terminal.

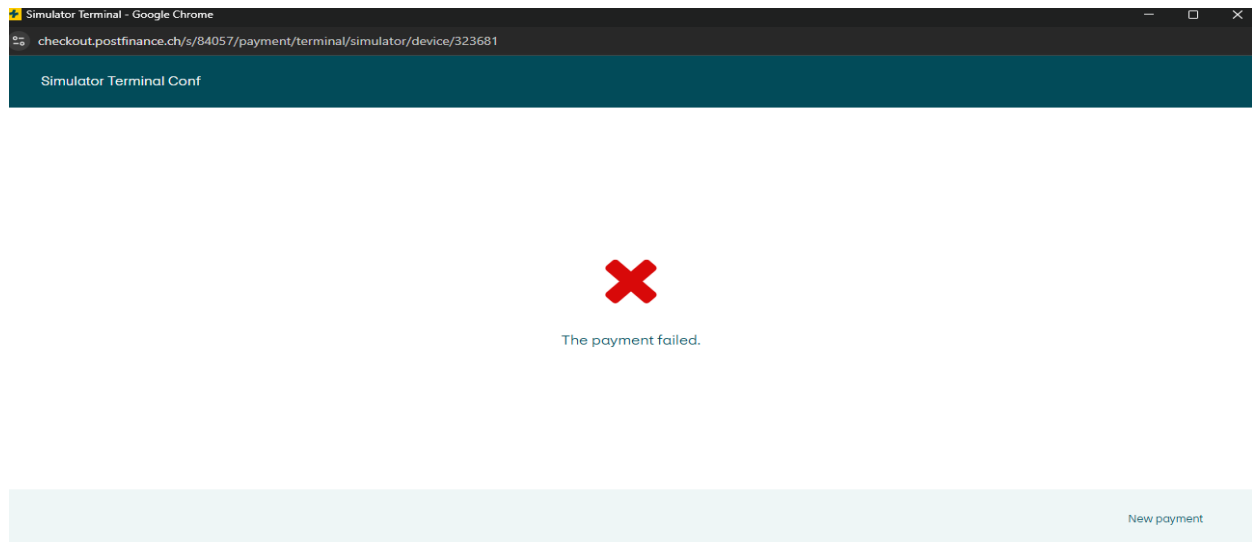


Figure 10: PostFinance Payment Cancelled Screen

Important: If a payment fails in Odoo but the transaction remains active on the PostFinance terminal, the cashier must cancel the transaction manually from the PostFinance terminal.

If the customer cancels the transaction from the PostFinance terminal, the payment is treated as cancelled/failed in Odoo.

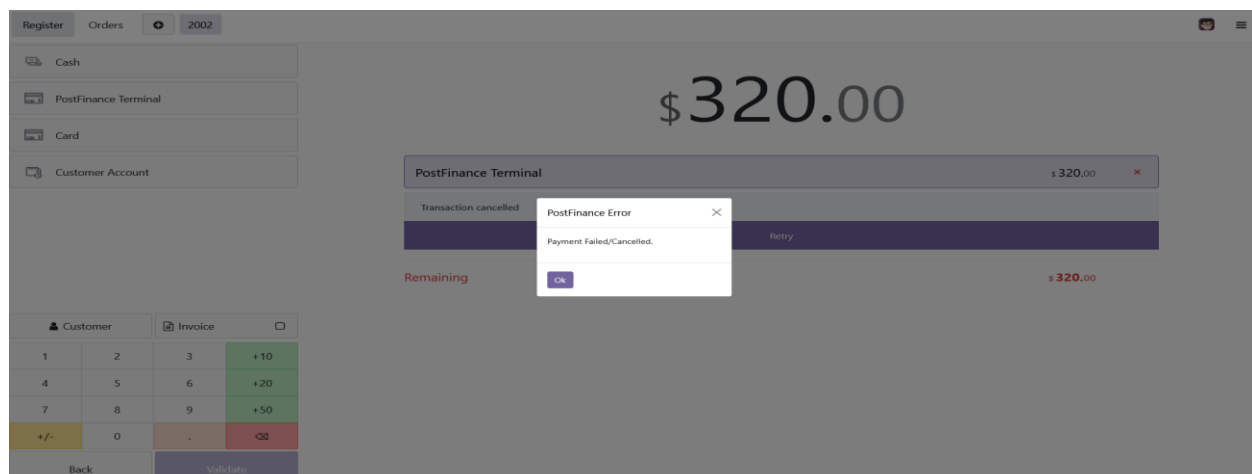


Figure 11: Odoo POS Payment Cancelled Screen

If a customer is associated with the order, a payment cancellation/failure email may be sent to the customer.

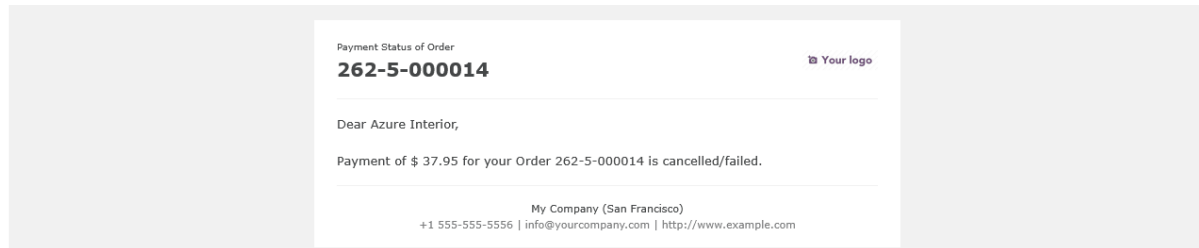


Figure 12: PostFinance Payment Cancelled/Failed Email to Customer

9 Refunds

The plugin supports refunds for POS orders paid through PostFinance.

There are two supported refund workflows:

1. Refund from the original POS order.
2. Refund from the POS Store.

9.1 Refund From the POS Order

Open the original POS order and click: **Return Products**

Orders
Test Shop 2 - 000013

Pickings
1

Invoice
Return Products
New
Paid
Posted
Send message
Lo

Order Ref ?
Test Shop 2 - 000013

Origin ?
Point of Sale

Date ?
Apr 2, 4:49 PM

Session ?
Test Shop 2/00022

User ?
Mitchell Admin

Customer ?

Fiscal Position ?

Products
Payments
Extra Info
Customer Note

Product	Lot/serial Number	Quantity	Unit Price	Disc.%	Taxes	Tax Excl.	Tax Incl.
[E-COM12] Conference Chair (Steel)		1.00	\$ 33.00	0.00	15%	\$ 33.00	\$ 37.95

Taxes ? : \$ 4.95
Total ? : \$ 37.95
Total Paid (with rounding) ? : \$ 37.95
Margin ? : \$ 33.00 (100%)

Figure 13: Refund option from the pos order

A refund POS order is created.

1. Set the quantity to be refunded.
2. Click **Payment**.
3. Select the PostFinance payment method.
4. Confirm the refund.

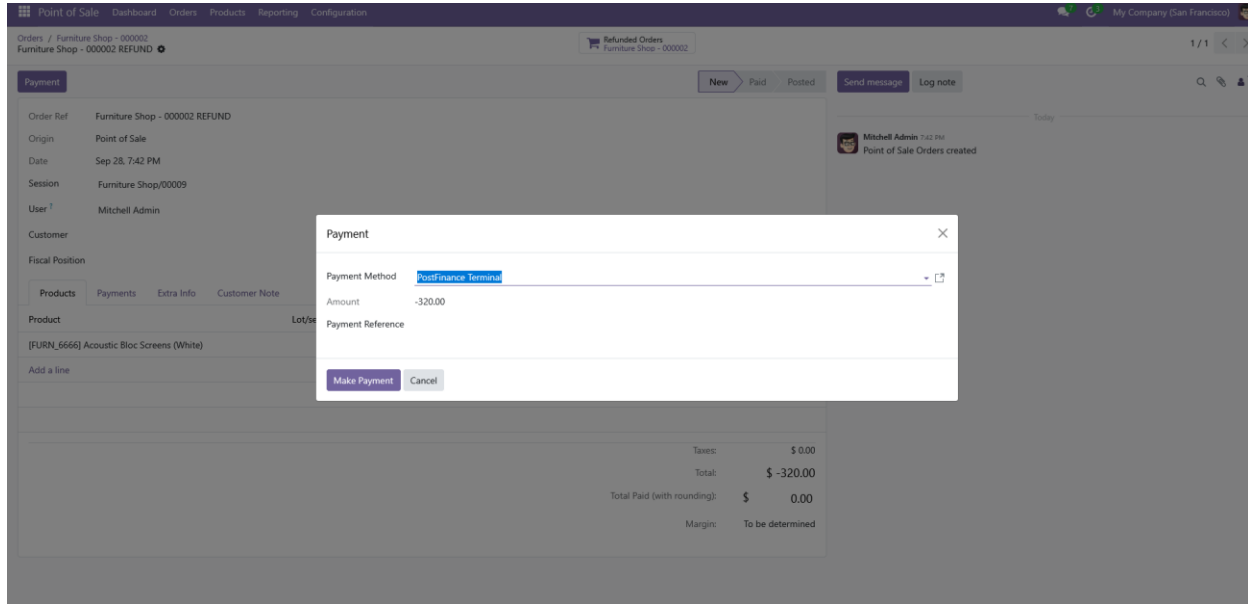


Figure 14: Payment Method Popup for Refund option from the pos order

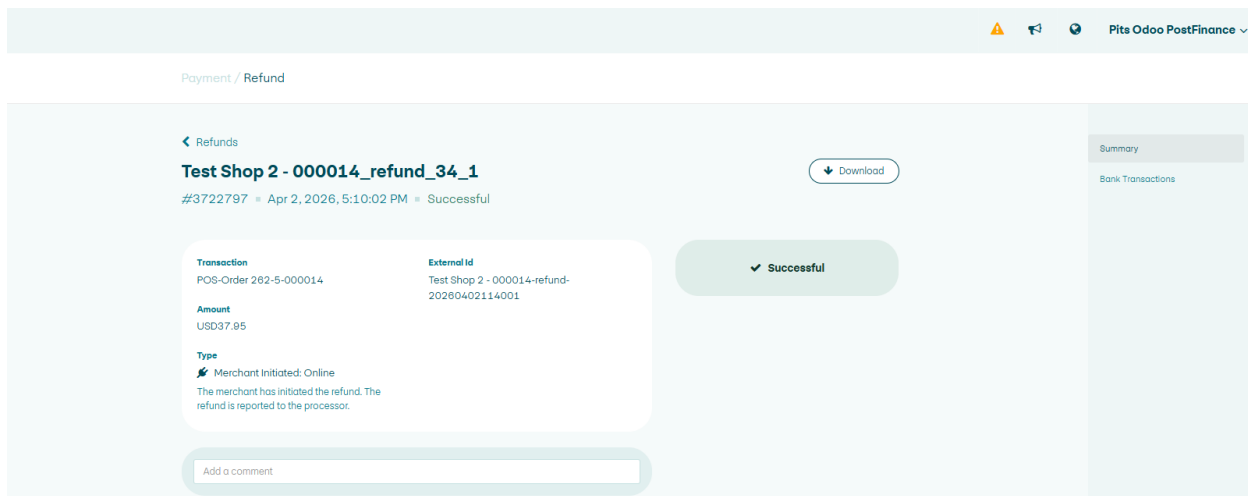


Figure 15: Refund in PostFinance Back Office

9.2 Refund From the POS Store

1. Go to the POS Store and select: **Actions** → **Refund**

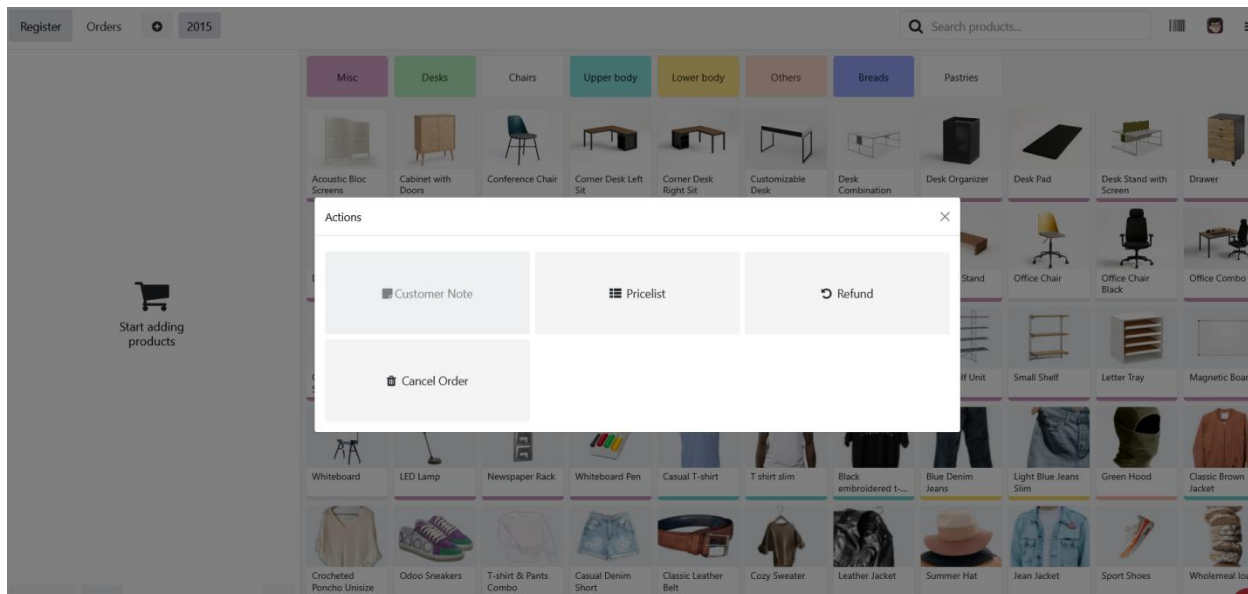


Figure 16: Refund option through POS Store

2. Select the order to be refunded.
3. Click **Refund**.

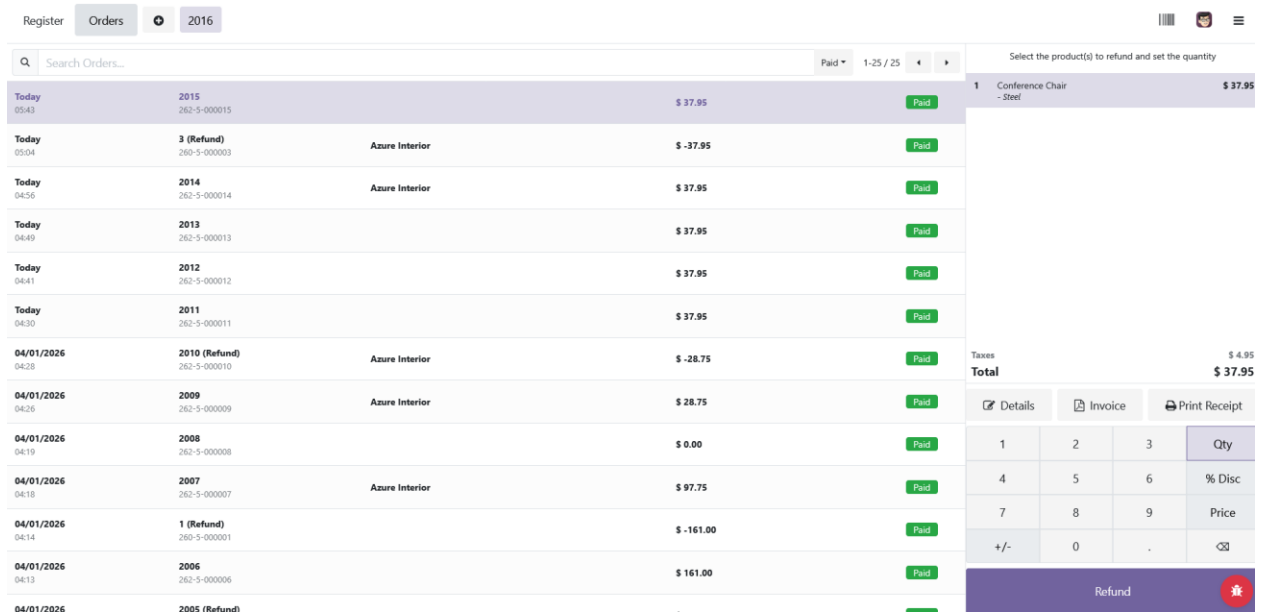


Figure 17: Refund option through POS Store

4. Select the PostFinance payment method and click **Refund**.

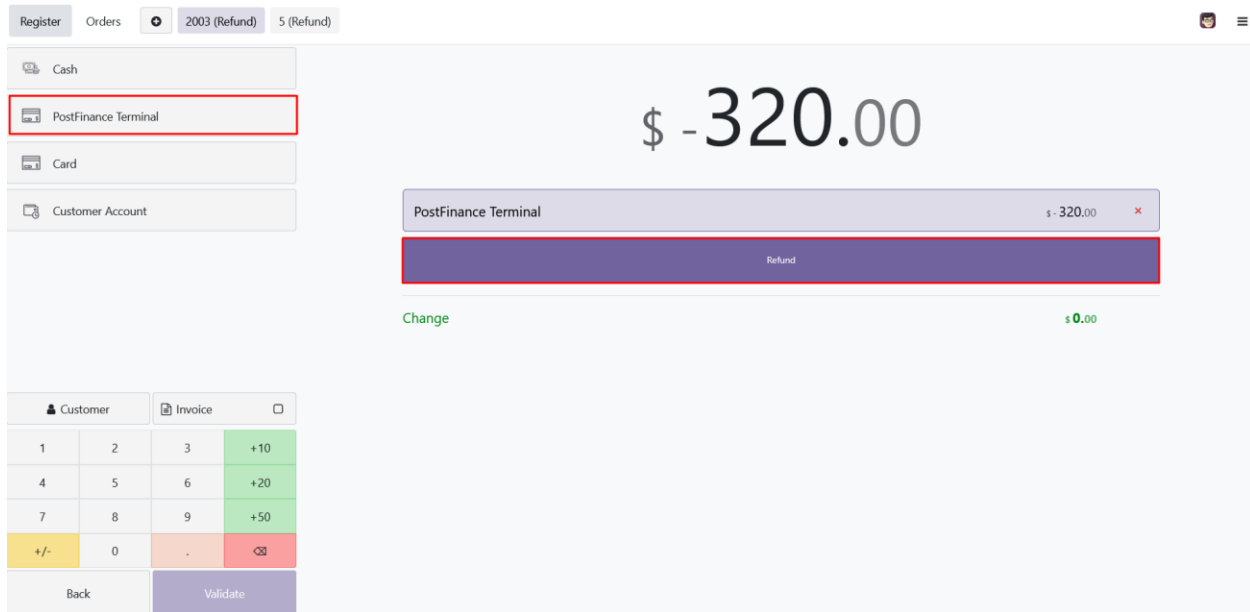


Figure 18: POS Store Refund Payment

The refund transaction is then processed automatically.

If a customer is selected for the POS order, an email will be sent to the customer after a successful refund. The email includes the refund amount and details of the refunded payment.

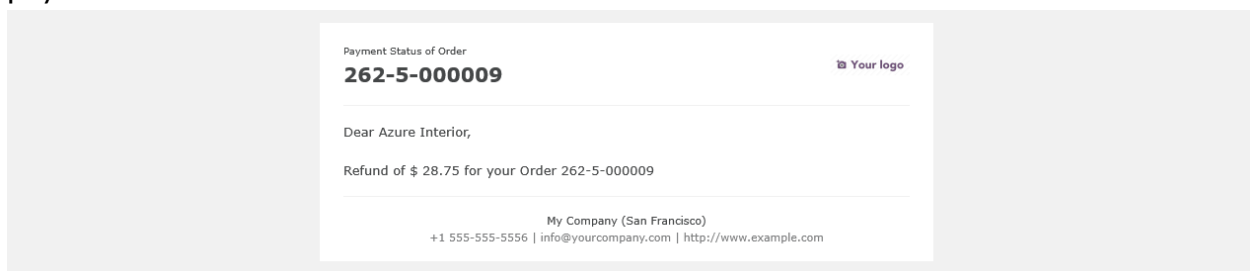


Figure 19: PostFinance Payment Refund Email to Customer

9.3 Refund Requirements

The original PostFinance transaction must be in the **Fulfilled** state before a refund can be processed.

When the customer initially makes a payment, the transaction may be in the **Authorized** state.

This means that the payment has been authorized but has not yet reached the required state for refund processing.

The transaction must be completed in the PostFinance Back Office and reach the **Fulfilled** state before the refund can be processed.

9.4 Refund Validation Messages

Transaction is still Authorized

If the original transaction is still in the Authorized state, the refund cannot be processed.

The system displays a message similar to:

PostFinance Error: Cannot refund transaction in state 'AUTHORIZED'. The transaction must be completed in the PostFinance Dashboard before a refund can be processed.

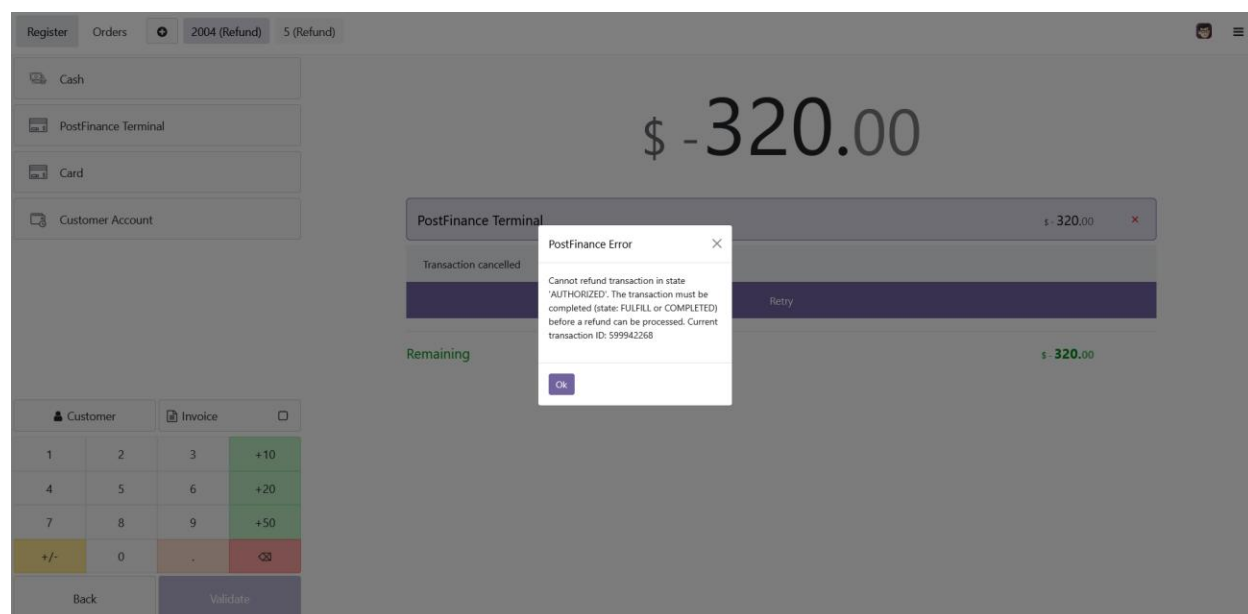


Figure 20: Validation for Authorized State in POS Store

Complete the transaction in the PostFinance Back Office and retry the refund.

Non-PostFinance Order

If a PostFinance refund is attempted for an order that does not contain a PostFinance transaction, Odoo displays:

No PostFinance transactions found for the original order.

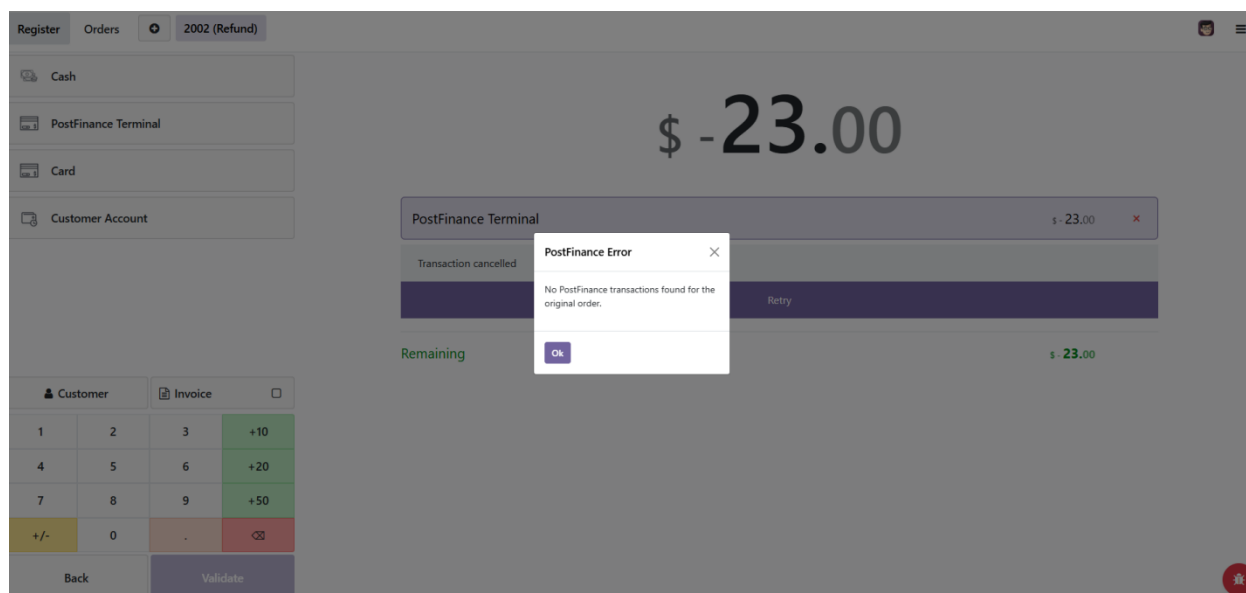


Figure 21: Validation for Selecting Non-PostFinance Payment Method

Different PostFinance Space

If the original payment was made using one PostFinance Space and the refund is attempted using a payment method configured with another Space ID, Odoo displays a validation error indicating that the order and selected PostFinance payment method belong to different Space IDs.

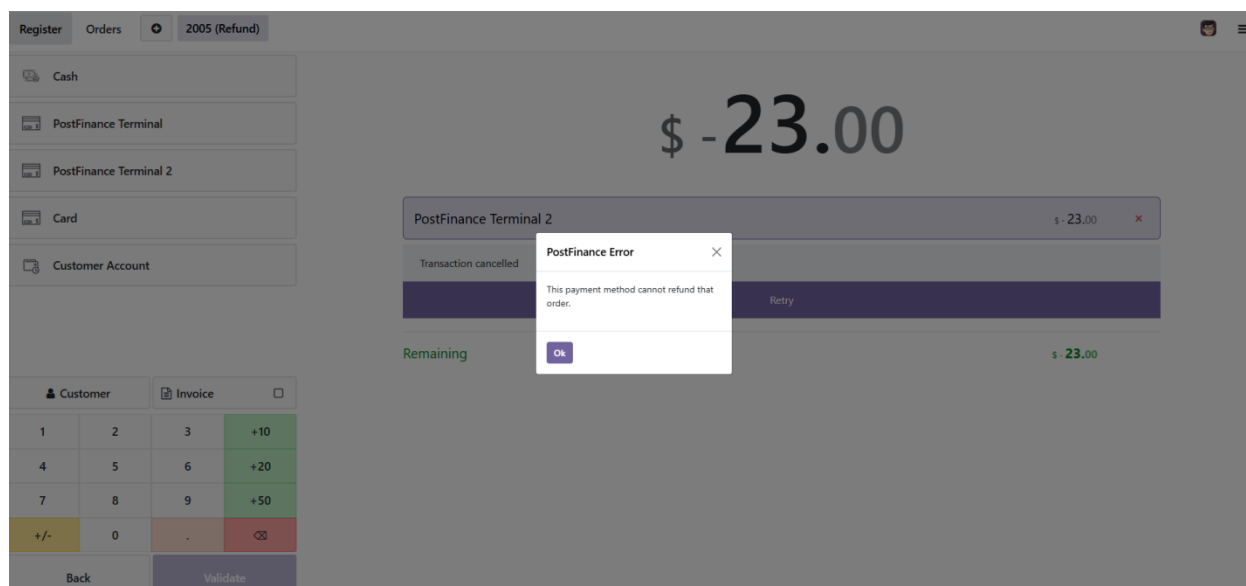


Figure 22: Validation for selecting PostFinance payment method in another Space ID

Use the PostFinance payment method configured for the same Space as the original transaction.

10 Transaction Information

After a successful payment, transaction information is available in:

Odoo

The POS order and related payment transaction can be viewed from the Odoo backend.

PostFinance

The corresponding transaction can be viewed in the PostFinance Back Office under the transaction records.

This allows the merchant to reconcile and track payment transactions between Odoo and PostFinance.

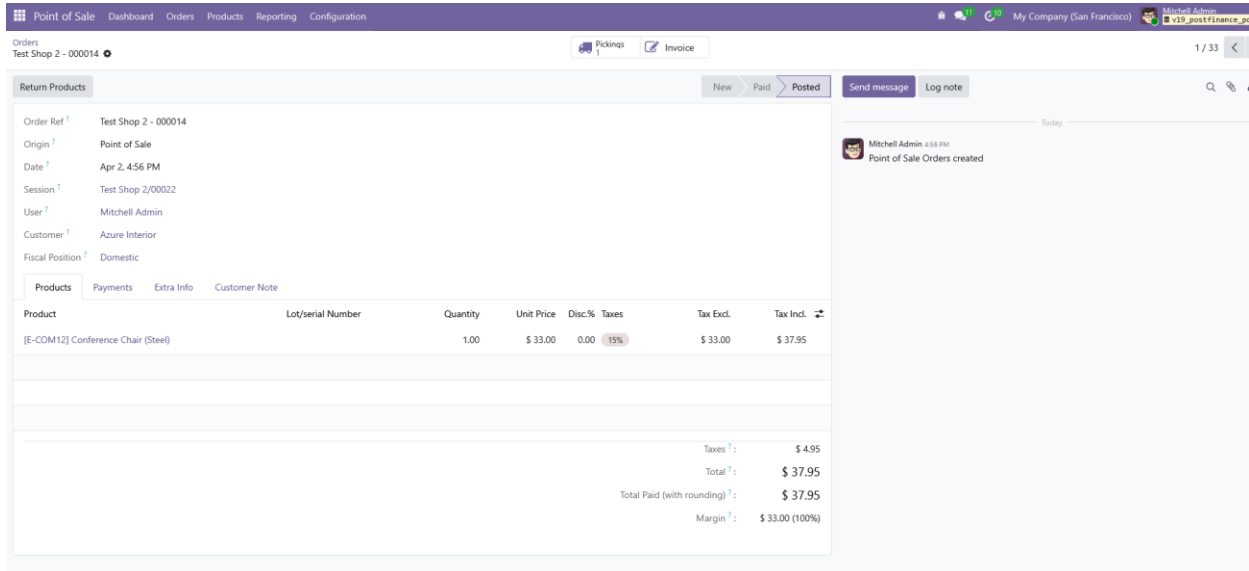


Figure 23: Odoo POS Order Screen

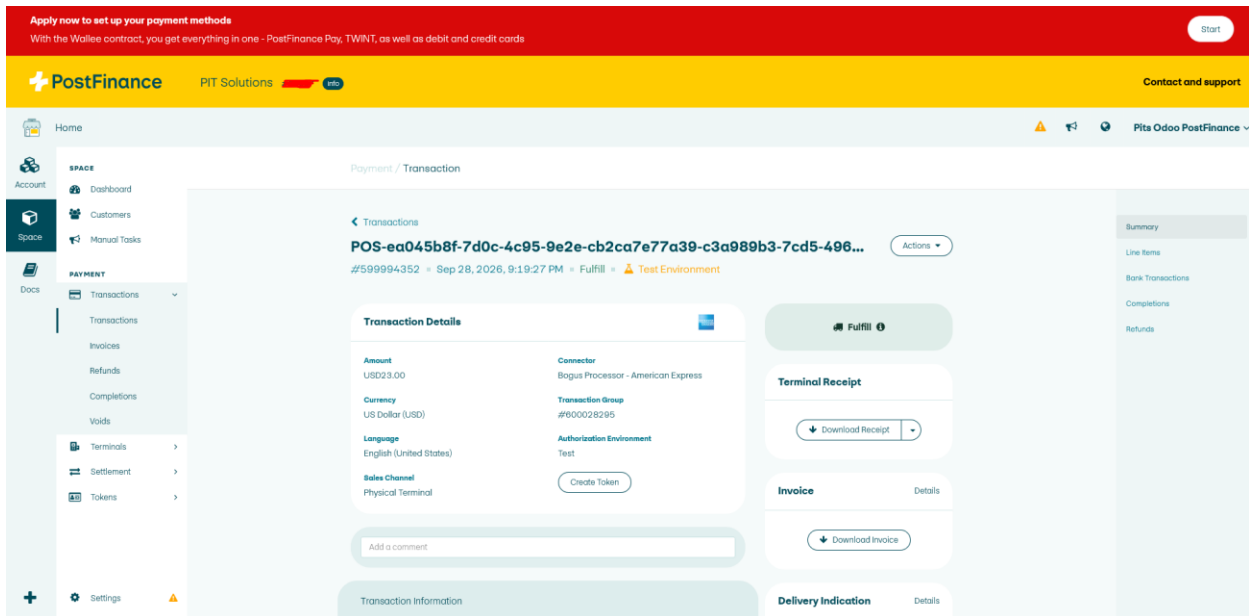


Figure 24: Payment Transaction in PostFinance Back Office

11 Troubleshooting

11.1 "External dependency not satisfied" / Module Not Installable

This normally indicates that one or more required Python dependencies are not available.

On-Premise

1. Activate the Python environment used by Odoo.
2. Install the plugin requirements:

```
pip install -r pits_pos_postfinance/requirements.txt
```

3. Restart Odoo.
4. Try installing the module again.

Odoo.sh

1. Verify that the required packages are listed in: *<Git repository root>/requirements.txt*
2. Commit and push the changes.
3. Wait for the Odoo.sh build to complete successfully.
4. If the build fails, review the Odoo.sh build and Python package installation logs.
5. After the build succeeds, update the Odoo Apps list and try installing the module again.

11.2 Payment Provider Cannot Be Saved

Verify that all four credentials have been entered:

- Space ID
- User ID
- Application Key
- Terminal Identifier

These fields are required when the provider state is **Test Mode** or **Enabled**.

11.3 Plugin Does Not Appear in Apps

Check the following:

- The *pits_pos_postfinance* module is present in the configured addons path.
- The module folder name has not been changed.
- The Odoo server/build contains the latest version of the module.
- The Apps list has been updated.
- The Odoo.sh build completed successfully, if using Odoo.sh.

11.4 PostFinance POS Option Does Not Appear

After configuring the payment method and assigning it to the POS shop:

1. Close the active POS session.
2. Start a new POS session.
3. Check the available payment methods again.

The updated terminal configuration is loaded when the new POS session starts.

12 Technical Requirements and Compatibility

- **Technical Requirements:** Requires Odoo Point of Sale, Payment, and Invoicing modules.
- **Compatible With:** Odoo 19.0 Community and Enterprise Editions.
- **Python Package Dependencies:** PostFinance Checkout SDK, CairoSVG, Pillow and Requests

13 Change Log / Release Notes

- Version 1.0.1: October 2026
 - Added python test scripts and minor enhancements.

- Version 1.0.0: September 2026
 - Initial release.

14 Support

If you have questions, use our contact form at webshopextension.com or email at support@webshopextension.com.