USER GUIDE · VERSION 1

Closed-loop e-wallet

Customer stored value, plus staff incentive balances with controlled payout.

1. What the e-wallet is for
2. Sign in and find your way
3. Look up someone's balance
4. Approve or reject a staff withdrawal
5. Check a customer's top-up
6. Trace a money movement in the journal
7. Find a customer's vouchers
8. Set up vouchers and give them to customers
9. Choose how customers can top up
10. Manage merchants and who can see what

What the e-wallet is for

For: Everyone

The e-wallet holds money that belongs to people but stays inside your company. "Closed-loop" means the money can only be used with your company: a customer tops up their wallet and spends it at your company's outlets, and it cannot be sent to another shop or another person. It holds two kinds of balance. **Customer balances** come from customers topping up in your company's customer app, and customers also keep their shopping vouchers here. **Staff incentive balances** come from the Staff incentive module, which works out what each person earned; staff then ask for that money to be paid into their bank account, and your finance team approves each request before it is paid.

Who uses it

- **Customers** top up, check their balance and use vouchers in the customer app. They never see the screens in this guide.
- **Staff** see their incentive balance and ask for a withdrawal in the Staff incentive app, not here.
- **Your finance team and administrators** use the wallet's admin screens, described in the rest of this guide: approving withdrawals, checking top-ups, looking up balances, and setting up vouchers and top-up options.

The admin screens have one built-in role, **Sysdev**, which can do everything. An administrator can add more roles under **Roles** (see the last page).

What it needs from other modules

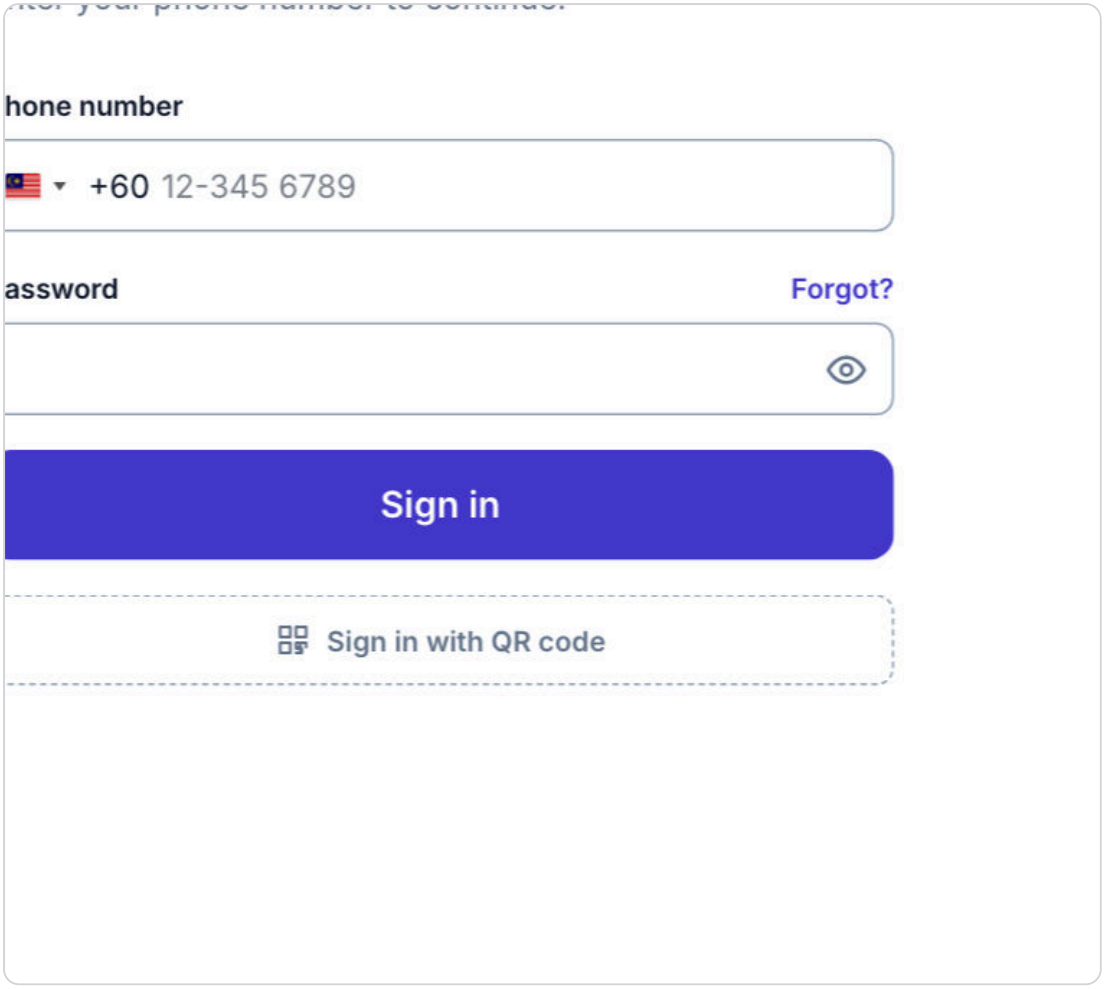
- **People's names** come from the app that opened each wallet. The wallet itself keeps only a phone number (staff) or a long sign-in code (customers), plus the name that app sent.
- **What staff earned** comes from the Staff incentive module. The wallet does not decide incentives.
- **Top-ups** go through your company's payment provider (shown on screen as UPG). The wallet never takes card or bank details itself.
- **Payouts to staff** go through your company's bank link, which your IT team sets up.

- **Using a voucher** is done by the customer app or the till talking to the wallet. There is no screen for it here.
- **Paying at an outlet from the customer's balance itself does not work yet.** This version of the wallet can take top-ups and hold the balance, but has no way for a till or the app to spend it. Ask your IT team when that is planned.

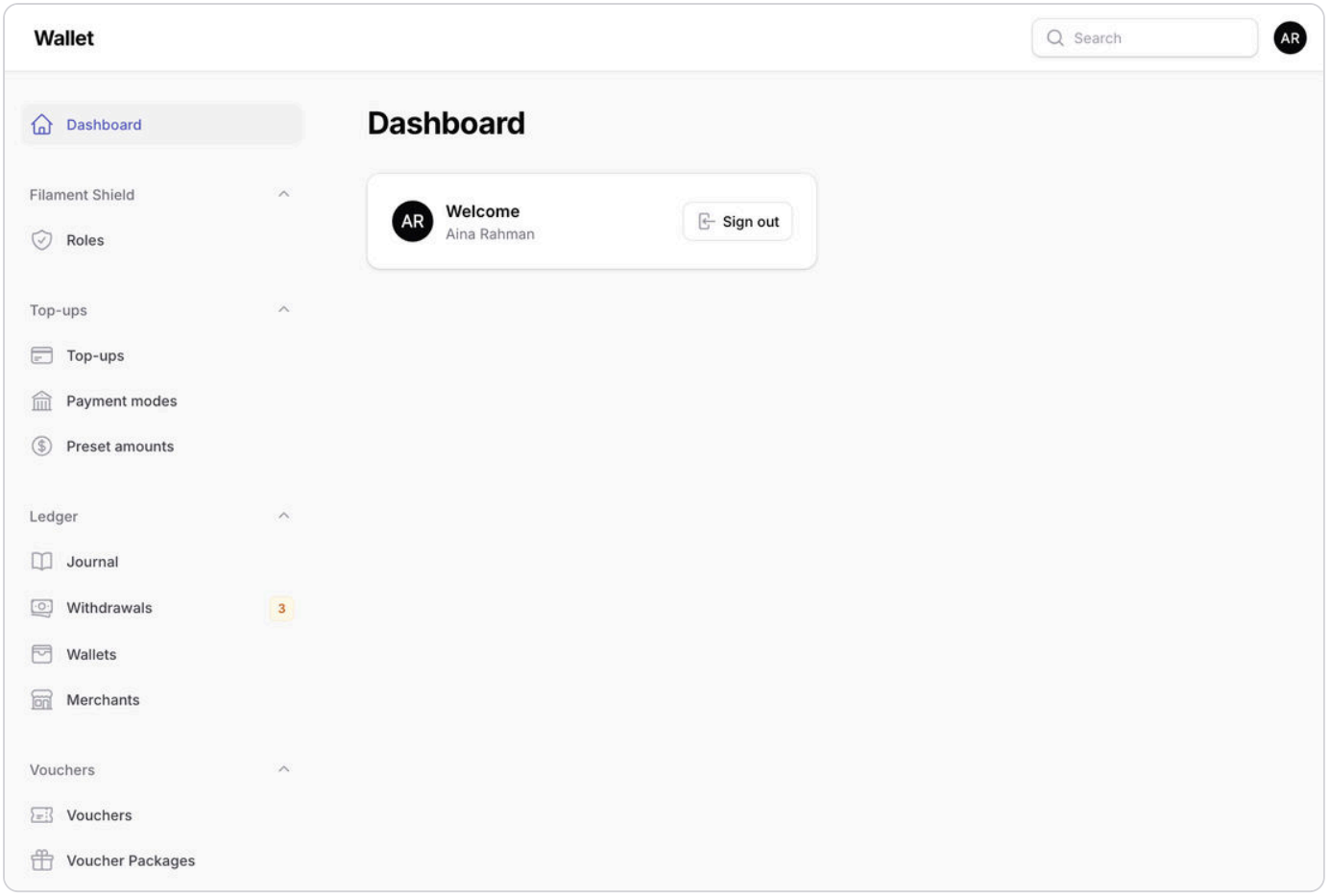
Sign in and find your way

For: Finance team and administrators

1. Open the wallet's web address that your IT team gave you. If your company uses the Portal, you can also open it from there.
2. On the **Sign in** page, type your **Phone number** and **Password**, then press **Sign in**. You can also press **Sign in with QR code** and scan the code with your phone.



3. You land on the **Dashboard**. The menu on the left holds everything:
    - **Top-ups**: customer top-ups, the **Payment modes** customers can pay with, and the **Preset amounts** the app offers.
    - **Ledger**: the **Journal** of every money movement, **Withdrawals** waiting for a decision, every **Wallet** and its balance, and the **Merchants** (the apps that use the wallet).
    - **Vouchers**: issued **Vouchers**, **Voucher Packages**, **Voucher Setups** and **Voucher Types**.
    - **Filament Shield**: **Roles**, which decide who can see the voucher screens.



The orange number next to **Withdrawals** is how many staff withdrawals are waiting for someone to approve or reject them.

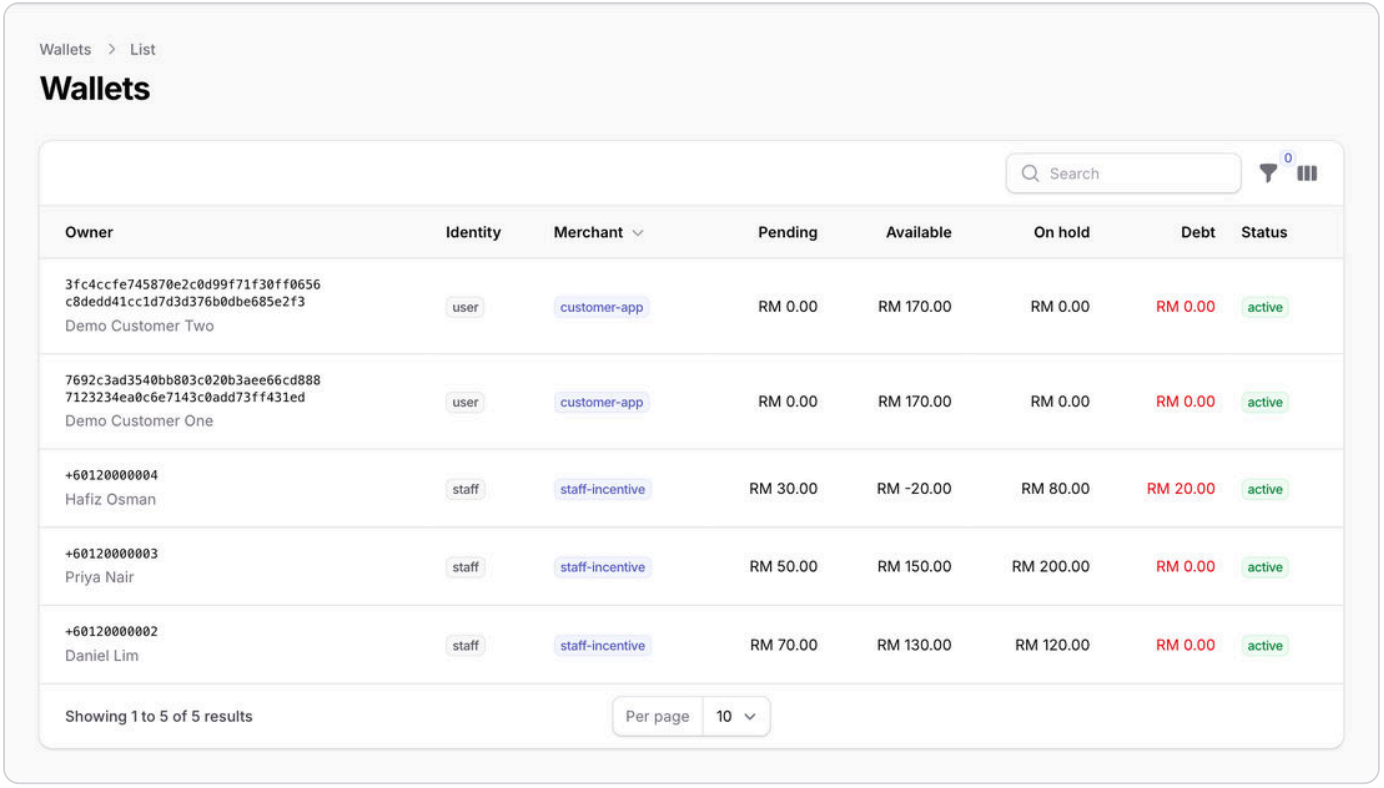
To leave, press **Sign out** on the Dashboard, or press your initials at the top right.

Look up someone's balance

For: Finance team and administrators

Use this when a customer or a staff member asks "how much do I have?" or "where is my money?".

1. Open **Wallets** in the menu, under **Ledger**.
2. Type the staff member's phone number into **Search**, for example 60120000002. Search does not look at names, so for a customer, scroll or filter until you see their name under the code. To see only one app's wallets, press the funnel button and pick a **Merchant**.



What the columns mean

- **Owner**: the staff member's phone number, or a long code for a customer, with their name underneath.
- **Identity**: **user** is a customer, **staff** is a staff member.
- **Merchant**: which app the wallet belongs to.
- **Pending**: staff incentive money that has been earned but is not ready to withdraw yet. Customer wallets show RM 0.00 here, because a top-up goes straight to Available.

- **Available**: money the person can use or withdraw now.
- **On hold**: money set aside for a withdrawal that has not been paid yet. It comes back to Available if the withdrawal is rejected or the bank sends it back.
- **Debt**: money the person owes. This happens when incentive money is taken back (for example, a sale was returned) after it was already paid out. Available can then show a minus figure, as in the Hafiz Osman row above. While someone has a debt they cannot withdraw, and their next earnings pay it off first.
- **Status**: **active**, or **locked** or **suspended** when the wallet has been stopped.

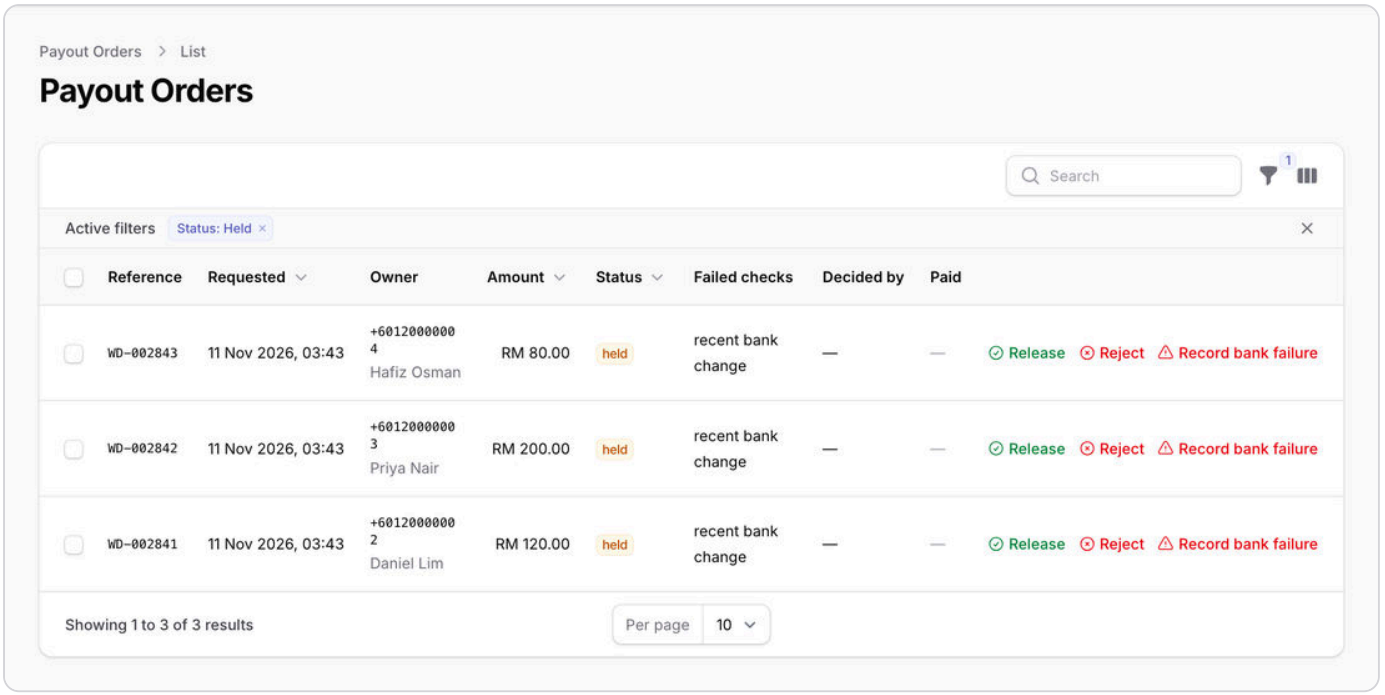
This screen only shows balances. You cannot change a balance here. Corrections are made by the app that owns the wallet, and each one is recorded in the **Journal** with a reason.

Approve or reject a staff withdrawal

For: Finance team and administrators

When a staff member asks for their incentive money in the Staff incentive app, the request comes here. Out of the box, no request is paid until a person approves it, whatever the amount.

1. Open **Withdrawals** in the menu, under **Ledger**. The page is headed **Payout Orders**, and it opens showing only the requests that are waiting (**Status: Held**).
2. Read each row: who asked (**Owner**), the **Amount**, and **Failed checks**.



Failed checks lists anything the system found worth a second look. It does not block the request; it is there to help you decide:

- **recent bank change**: the bank account was added or changed in the last 30 days.
- **repeat withdrawal**: the person already asked for a withdrawal in the last 24 hours.
- **auto release ceiling**: the amount is above the limit that could ever be paid without a person.
- **earnings spike**: recent earnings are much higher than this person's usual.
- **fraud review**: the Staff incentive module says an earning is being checked for fraud.
- **recoverable debit**: the person owes money (see **Debt** on the Wallets page).

To approve

3. Press **Release** on the row.
4. Type why in **Why is this being released?** (for example "Checked against the monthly incentive list") and press **Submit**. The reason is kept with the withdrawal for good.

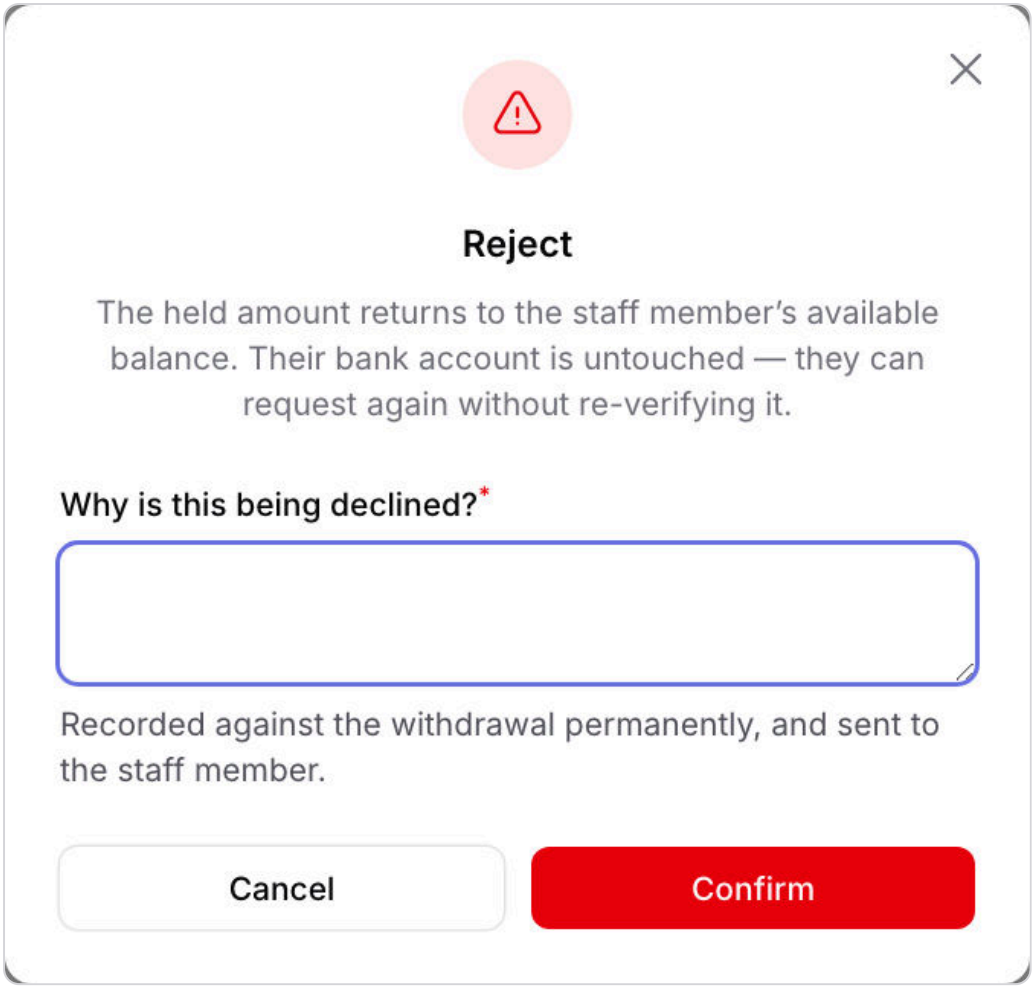


The payment then goes to the bank through your company's bank link. Its status moves to **released**, then **submitted**, then **paid** when the bank confirms. How quickly depends on how your IT team set up the bank link.

To approve many at once, tick the boxes on the left of the rows, then press **Release selected**. You type one reason, and it is recorded against each withdrawal. Any request that can no longer be released is left alone and listed in a message.

To reject

3. Press **Reject** on the row.
4. Type why in **Why is this being declined?** and press **Confirm**. The reason is kept with the withdrawal and sent to the staff member.



The money goes back to the person's **Available** balance. Their bank account stays approved, so they can ask again straight away.

When the bank sends a payment back

Use **Record bank failure** only when the bank returned or refused a payment, not to turn a request down. Type what the bank said and confirm. The money goes back to the person's **Available** balance, and their bank account has to be checked again before the next withdrawal.

To see older requests, press the funnel button and change **Status** (for example to **Paid** or **Rejected**). **Decided by** shows who released or rejected each one.

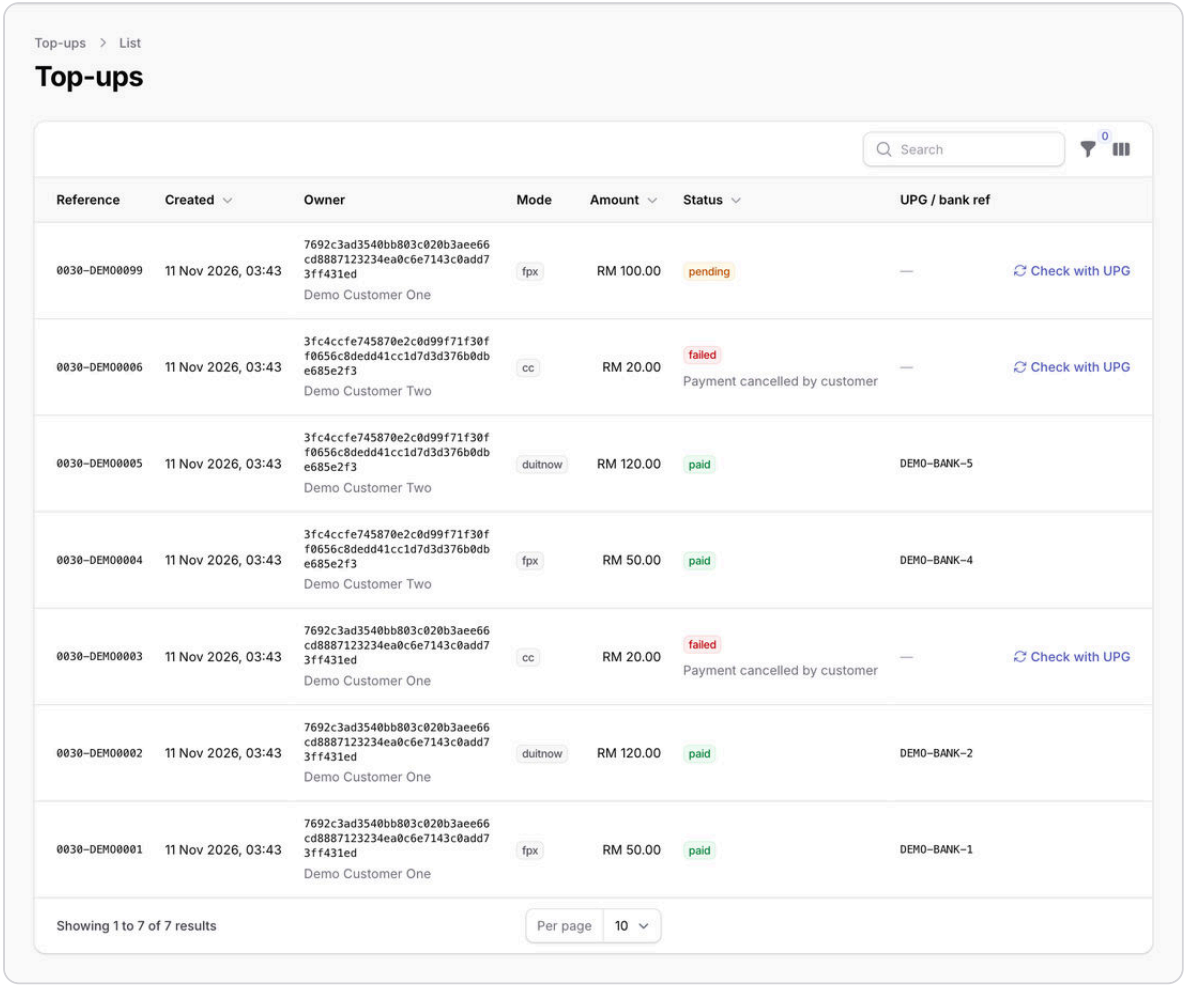
Check a customer's top-up

For: Finance team and administrators

Customers top up in the customer app: they choose an amount and a payment mode, and the app sends them to your company's payment provider (UPG) to pay. That payment step happens on the provider's own page, not in the wallet. When the provider confirms the payment, the money appears in the customer's **Available** balance straight away.

Use this screen when a customer says "I paid but my balance did not change".

1. Open **Top-ups** in the menu, under **Top-ups**.
2. Type the reference the customer quotes into **Search** (it starts with 0030-). Search does not look at names.



3. Read the **Status**:
    - **paid**: the money is in the customer's balance. **UPG / bank ref** is the provider's reference, if the customer needs to quote it to their bank.

- **pending**: the customer started paying but the provider has not confirmed yet.
- **failed**: the payment did not go through. The reason is shown under the status.
- **expired**: nobody finished paying in time.

4. For a **pending** or **failed** top-up, press **Check with UPG**. The wallet asks the payment provider for the latest answer. If the provider says it was paid, the money is added to the balance then. A message tells you the result.

The wallet also checks unfinished top-ups with the provider by itself every five minutes.

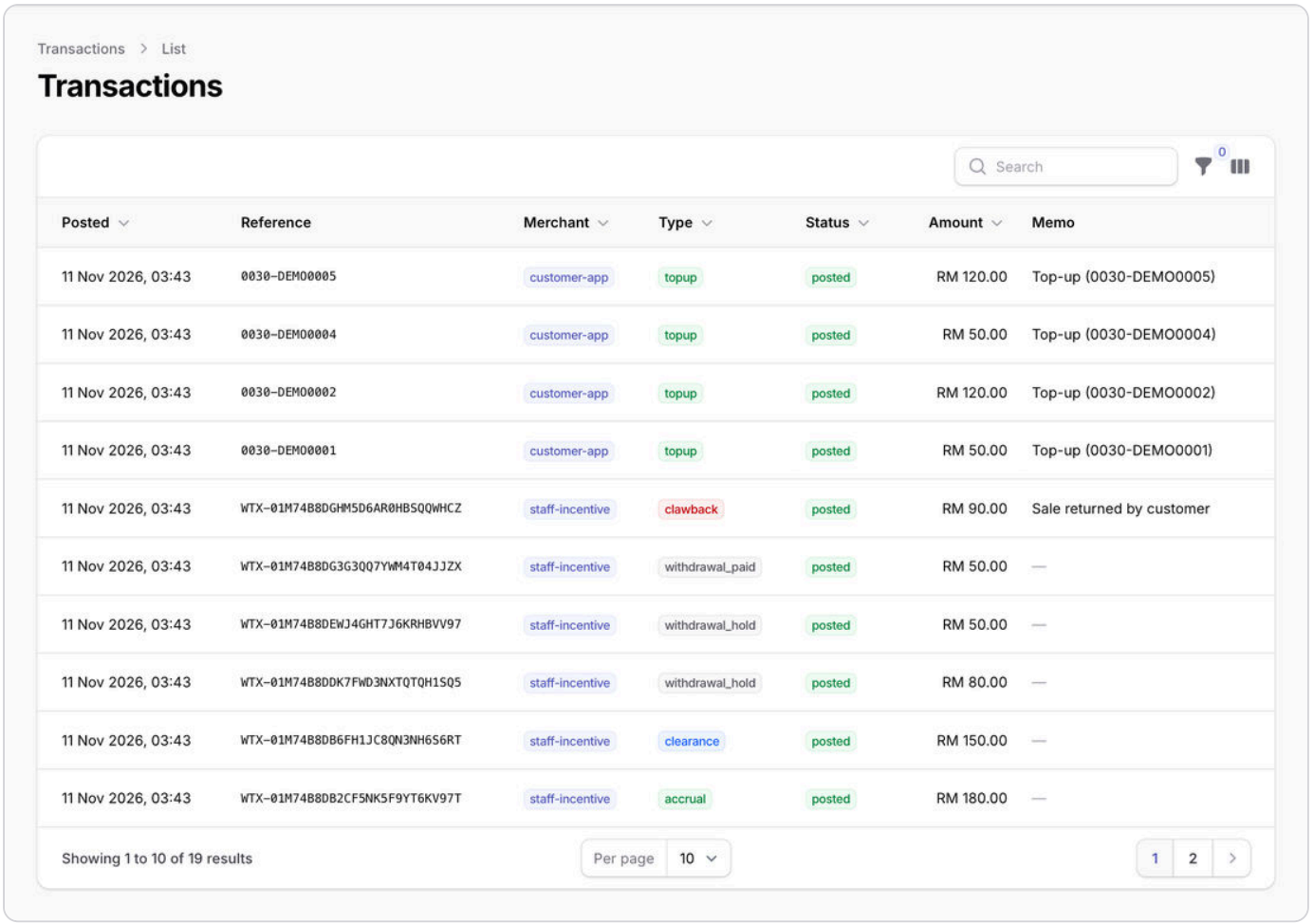
You cannot add money to a customer's balance from this screen. A top-up only counts once the payment provider confirms it.

Trace a money movement in the journal

For: Finance team and administrators

The **Journal** lists every movement of money in the wallet, newest first. Nothing in it can be edited or deleted: a mistake is fixed by adding a new movement that undoes it, so you can always see both.

1. Open **Journal** in the menu, under **Ledger**. The page is headed **Transactions**.
2. Search for a reference, or press the funnel button to filter by **Type**, **Status** or **Merchant**.



What the types mean

- **accrual**: a staff member earned incentive money. It starts in **Pending**.
- **clearance**: earned money became ready to withdraw (moved from Pending to Available).
- **clawback**: earned money was taken back, for example because a sale was returned. The **Memo** says why.

- **debit_recovery**: new earnings paid off money the person owed.
- **withdrawal_hold**: money set aside when a staff member asked for a withdrawal.
- **withdrawal_paid**: a withdrawal was paid to the bank.
- **withdrawal_returned**: a withdrawal came back (rejected, or refused by the bank) and the money went back to Available.
- **reversal**, **manual_adjustment**, **forfeiture**: corrections made by a person, each with a recorded reason. A forfeiture writes off the unclaimed balance of someone who has left.
- **topup**: a customer top-up that was paid.

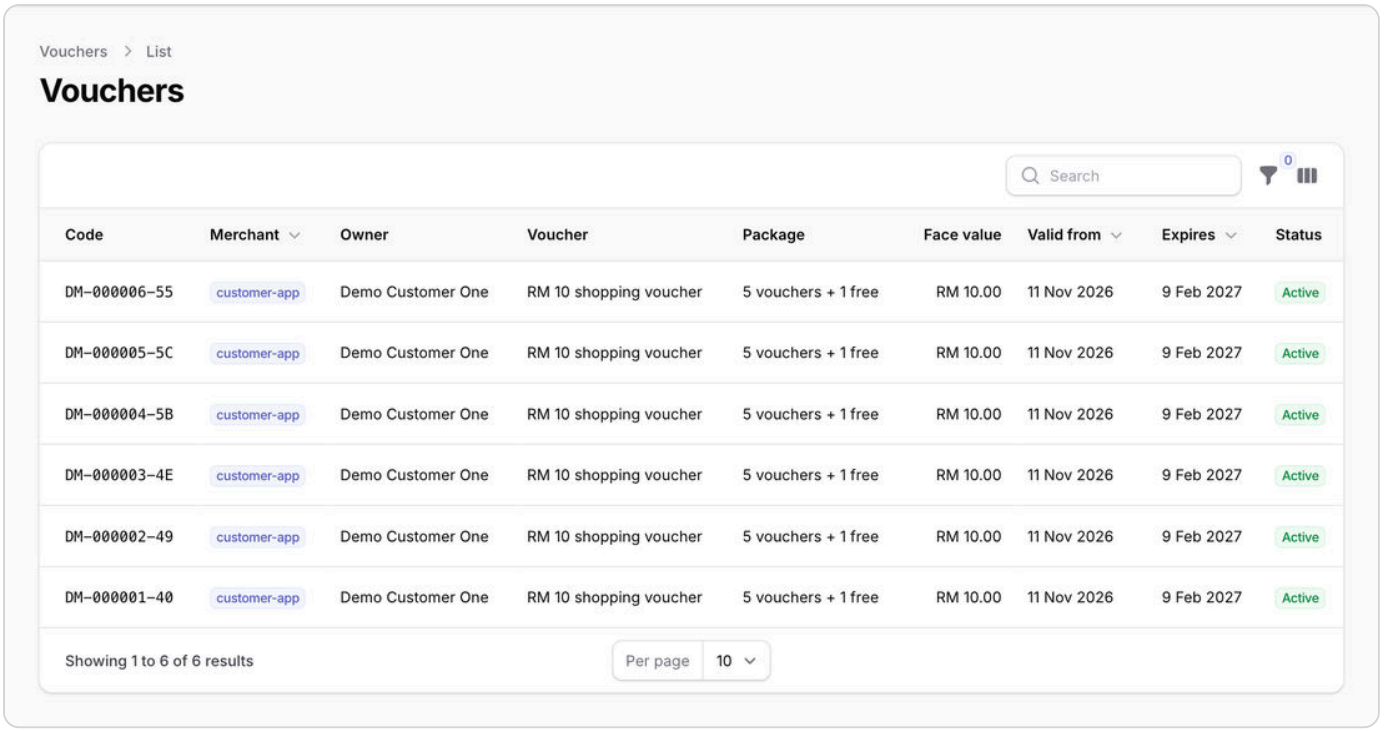
Corrections are made by the app that owns the wallet. There is no button for them on this screen.

Find a customer's vouchers

For: Finance team and administrators

Use this when a customer asks about a voucher: whether they have it, whether it is still valid, or whether it was used.

1. Open **Vouchers** in the menu, under **Vouchers**.
2. Type the voucher code into **Search**. You can type it with or without the dashes. Search also finds the voucher name, the package name and the reference typed when the vouchers were given out, but not the customer's name.



3. Check the row:
    - **Face value**: what the voucher is worth.
    - **Valid from** and **Expires**: the dates it can be used between.
    - **Status**: **Active** can still be used, **Redeemed** has been used, **Cancelled** has been cancelled.
    - **Package**: the bundle it came in. If **Owner** has "from" under it, someone else gave the voucher to this customer.

Press the columns button at the top right of the list to show more, such as **Paid** (what the customer paid for that voucher), **Free** and **Issued**.

Customers use their vouchers in the customer app or at the till. You cannot mark a voucher as used or cancelled from this screen.

You need a role with permission to see vouchers. If **Vouchers** is missing from your menu, ask an administrator.

Set up vouchers and give them to customers

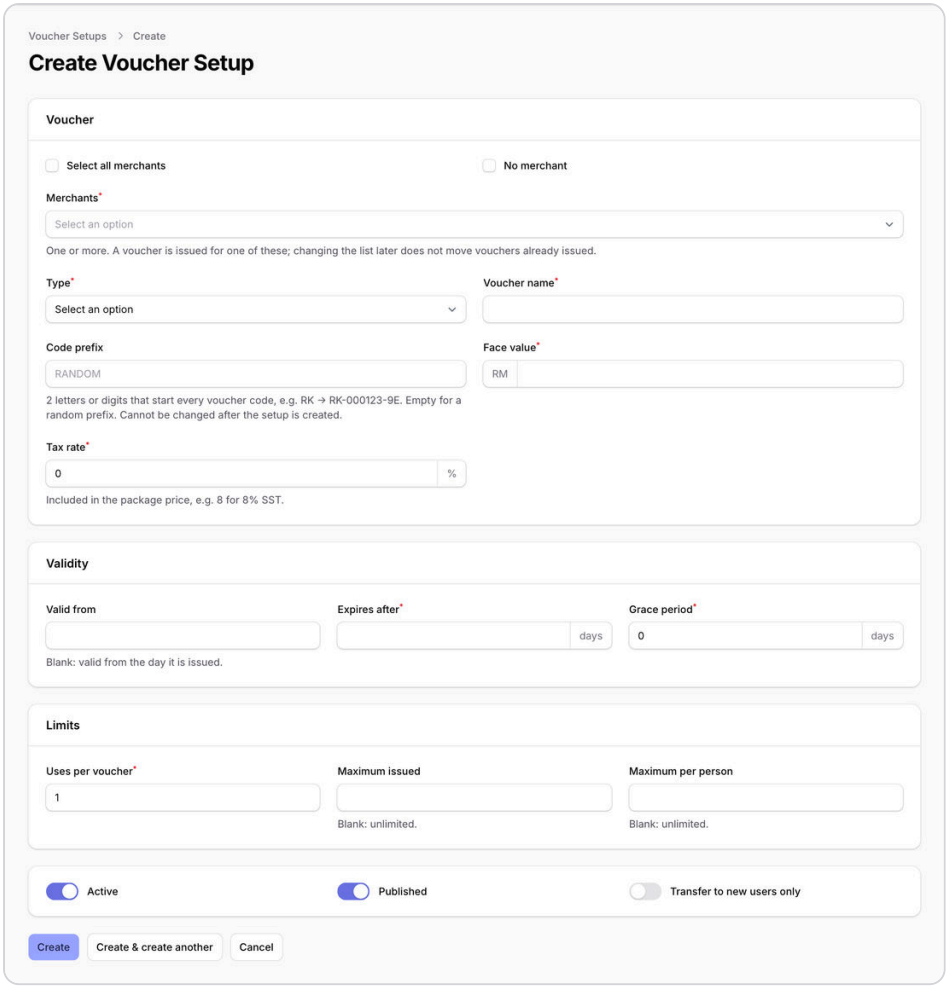
For: Administrators

Vouchers are built in three layers:

- A **voucher type** is the kind of voucher, such as **Cash Voucher** or **Percentage Discount**. Five types come built in.
- A **voucher setup** is one voucher design: its name, value and how long it lasts.
- A **voucher package** is what a customer buys or is given, for example "5 vouchers + 1 free" of one setup, at a price.

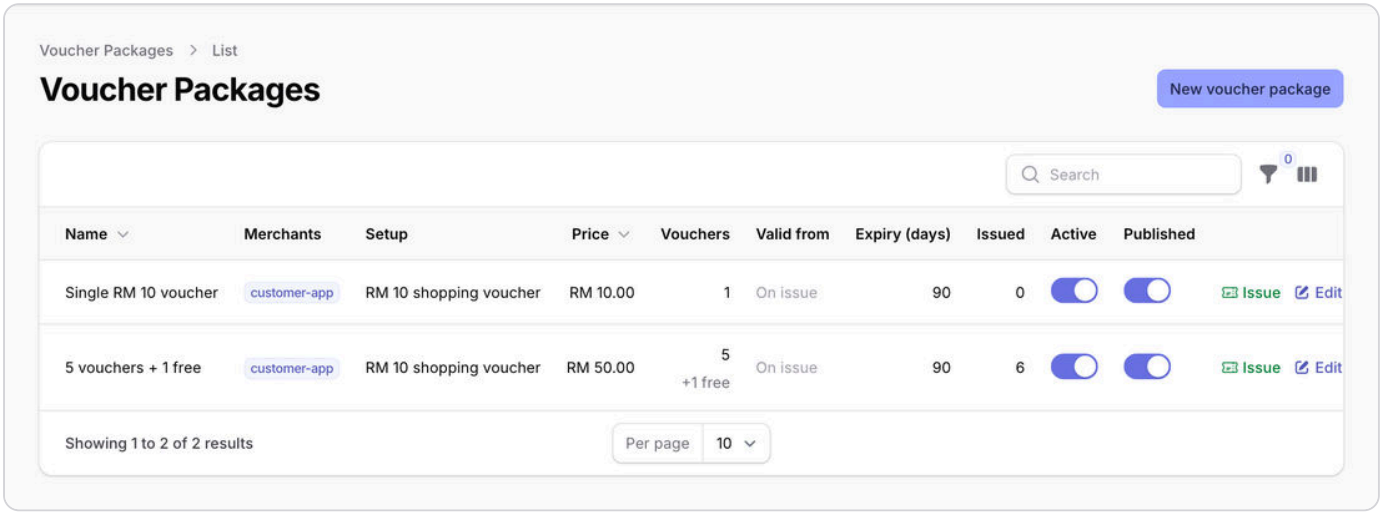
Make a voucher setup

1. Open **Voucher Setups** in the menu and press **New voucher setup**.
2. Under **Voucher**, pick the **Merchants** (apps) it can be used through, or tick **Select all merchants**. Pick a **Type** and type a **Voucher name**.
3. Fill in **Face value** (what one voucher is worth) and **Tax rate** (0 if none). **Code prefix** is optional: two letters or digits that start every code, for example DM gives codes like DM-000001-40. You cannot change the prefix or the type later.
4. Under **Validity**, fill in **Expires after** (days). Leave **Valid from** empty to start on the day each voucher is given out.
5. Under **Limits**, set **Uses per voucher**, and if you want a cap, **Maximum issued** and **Maximum per person**. Empty means no limit.
6. Leave **Active** and **Published** on, then press **Create**.



Make a package

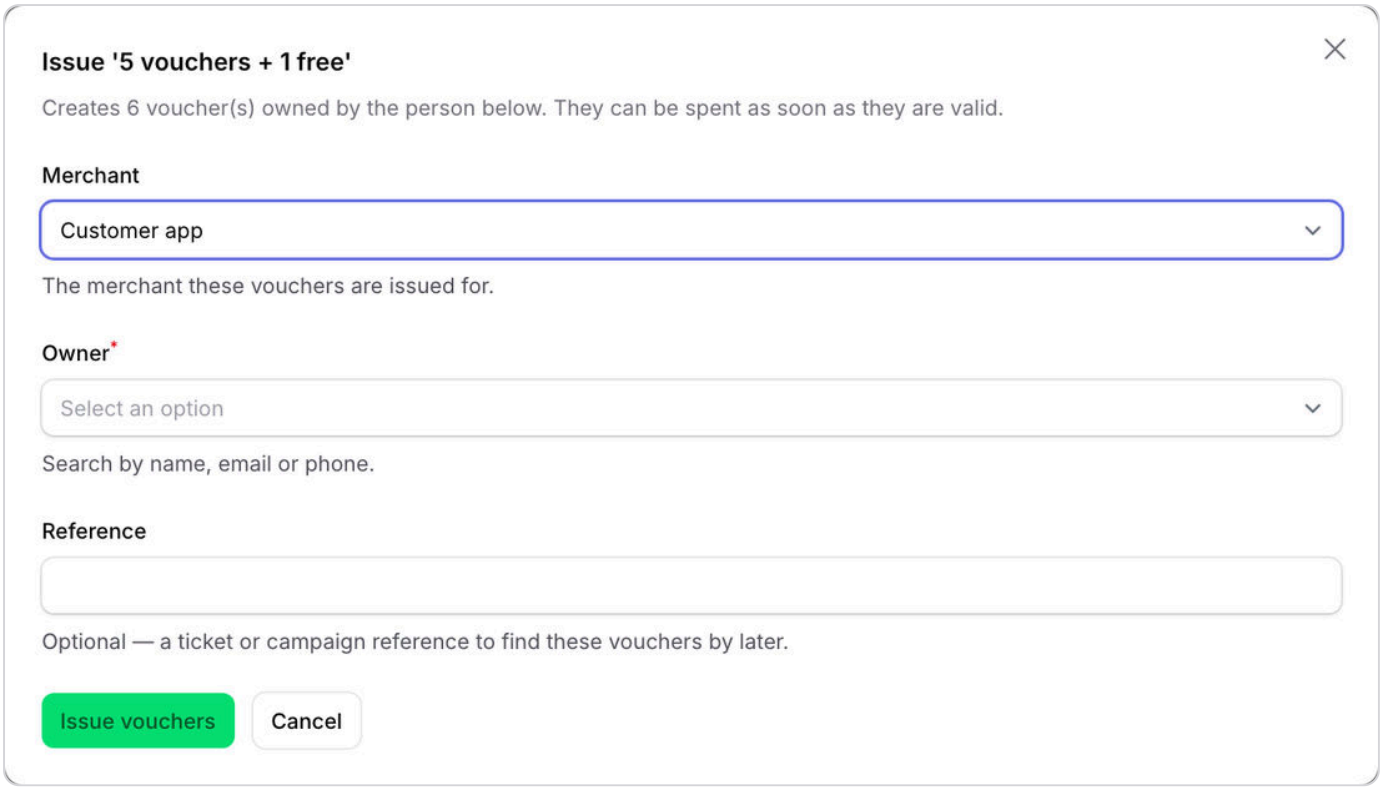
7. Open **Voucher Packages** and press **New voucher package**.
8. Pick the **Voucher setup**, type a **Package name**, the number of **Vouchers**, any **Free vouchers**, and the **Price** (including tax). The validity is copied from the setup; change it here if this package is different. Press **Create**.



Give a package to a customer

Use this for goodwill, a campaign, or to fix a problem, when the customer is not paying.

9. On **Voucher Packages**, press **Issue** on the package. You need a role that is allowed to issue vouchers.
10. Check the **Merchant**, then in **Owner** search for the customer by name, email or phone. Optionally type a **Reference** such as a campaign name, so you can find these vouchers later.
11. Press **Issue vouchers**. The customer gets every voucher in the package, free ones included, and can use them as soon as they are valid.



Only customers who have already signed in to the customer app at least once can be found in **Owner**.

To stop a package being sold or given, open it with **Edit** and switch off **Active**. **Published** controls whether the customer app shows it.

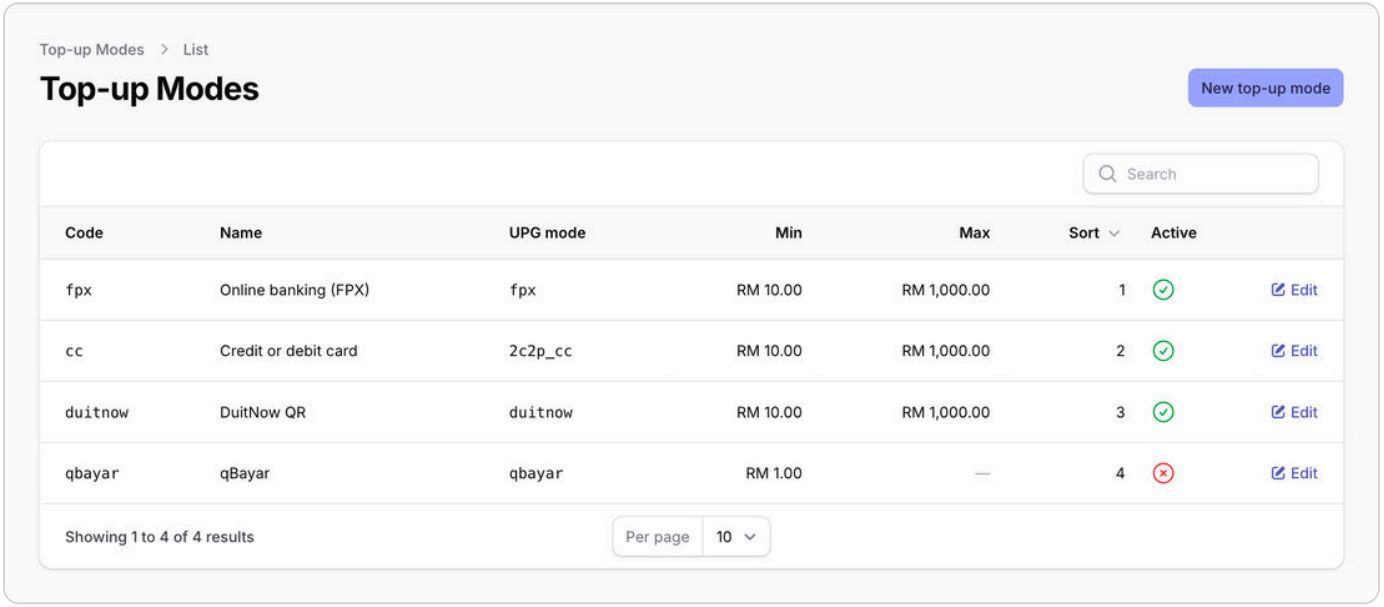
Choose how customers can top up

For: Administrators

These two screens decide what the customer app offers when a customer tops up.

Payment modes

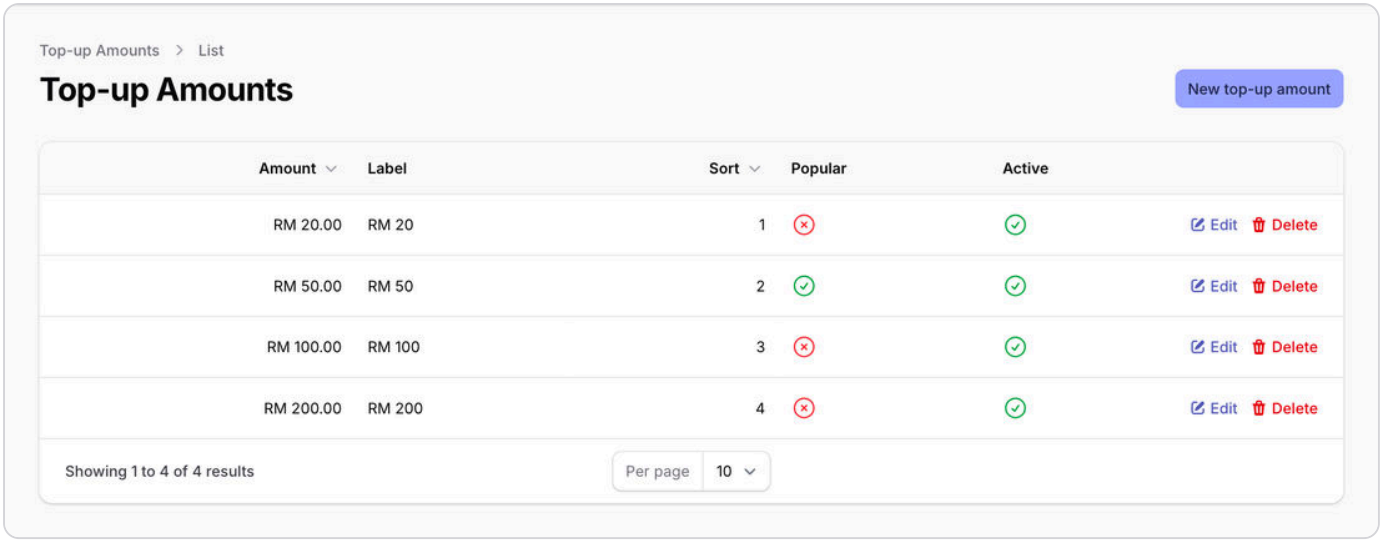
1. Open **Payment modes** in the menu, under **Top-ups**. The page is headed **Top-up Modes**.
2. Press **Edit** on a mode to change it, or **New top-up mode** to add one.



- **Name** is what customers see, for example "Online banking (FPX)".
- **Minimum** and **Maximum** are the smallest and largest top-up allowed with that mode. Leave **Maximum** empty for no limit.
- **Sort** sets the order in the app, lowest first.
- Switch off **Active** to hide a mode without deleting it.
- **Code** and **UPG payment mode** must match what the customer app and the payment provider expect. Ask your IT team before changing them; a wrong value stops that payment mode working.

Preset amounts

3. Open **Preset amounts** in the menu. The page is headed **Top-up Amounts**.
4. Press **New top-up amount** to add a quick-pick amount, or **Edit** or **Delete** on an existing one.



- **Label** is shown instead of the amount if you fill it in, for example "RM 50".
- **Popular** highlights the amount in the app.
- Switch off **Active** to hide an amount for now.

Customers can still type their own amount between the payment mode's minimum and maximum.

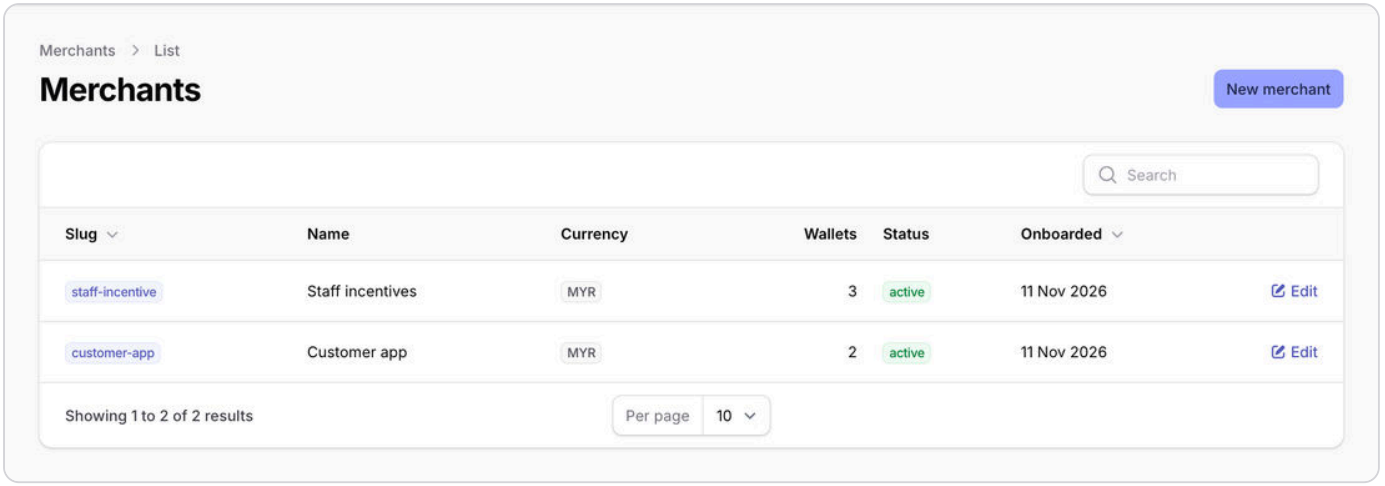
Manage merchants and who can see what

For: Administrators

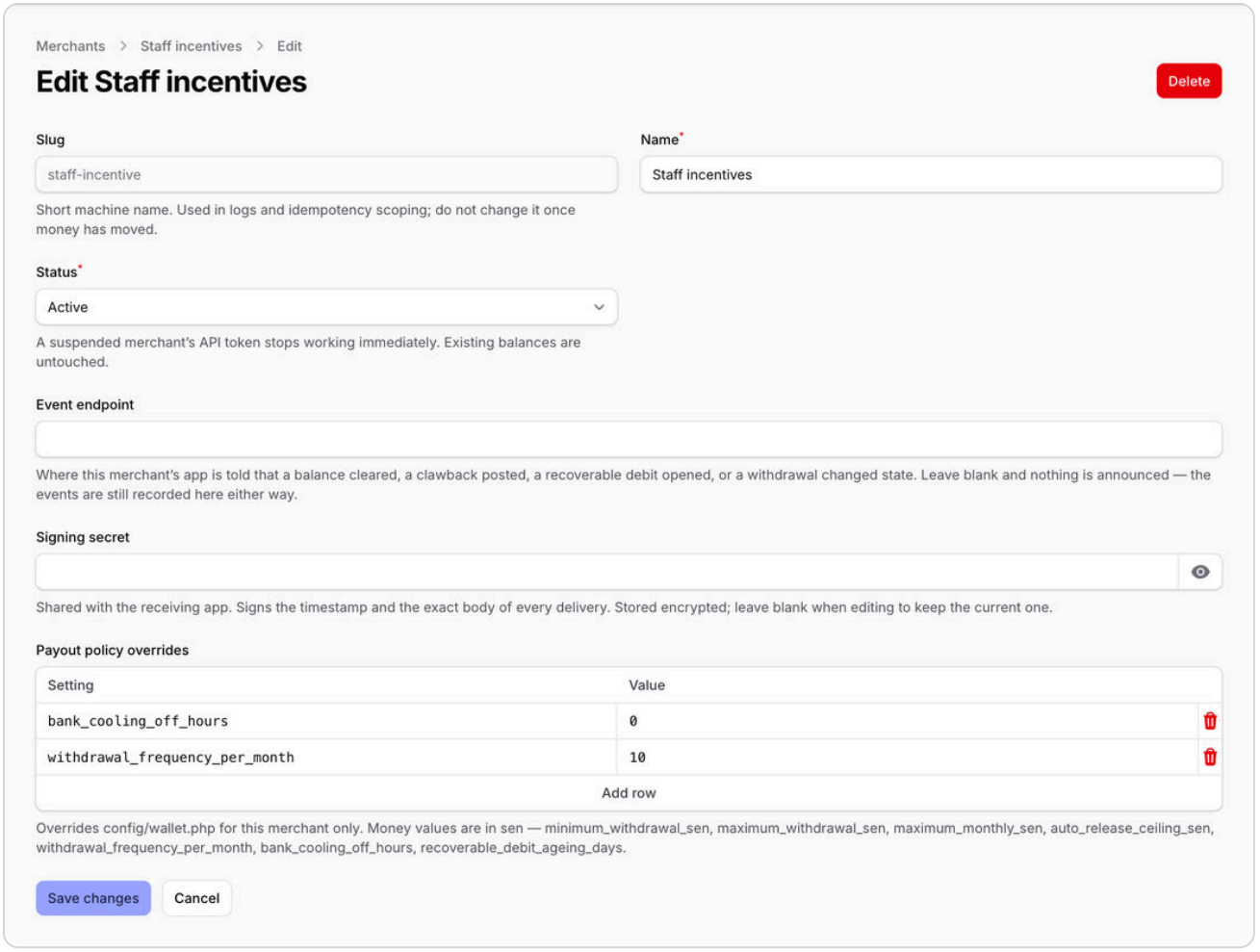
Merchants

A merchant here is one of your company's apps that keeps money in the wallet, for example the customer app and the Staff incentive module. Each app can only see and move its own wallets.

1. Open **Merchants** in the menu, under **Ledger**.



2. Press **Edit** on a merchant to change it.



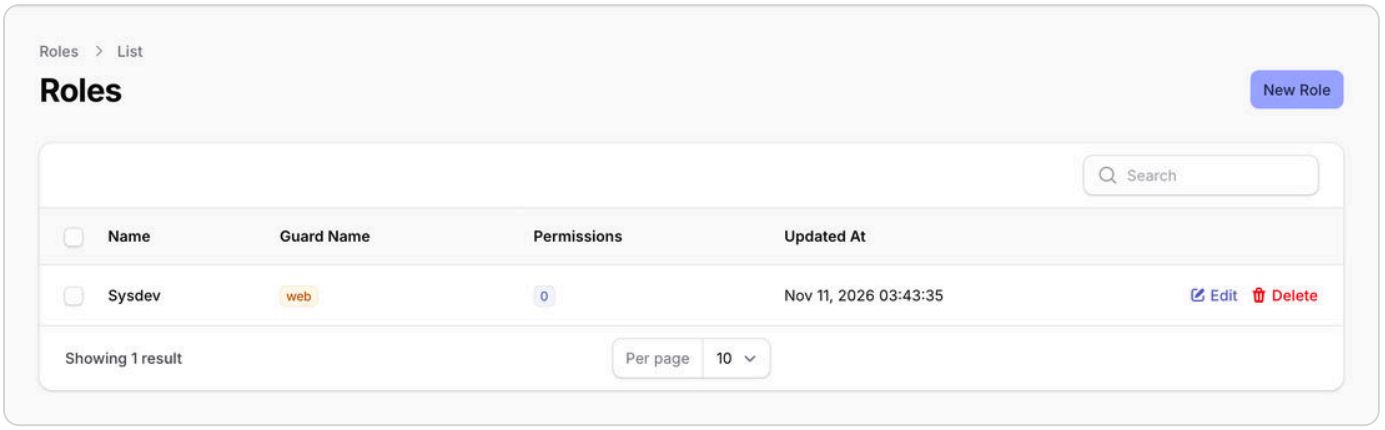
- **Status**: choose **Suspended** to stop that app using the wallet at once, for example if its password may have leaked. Balances stay as they are.
- **Event endpoint** and **Signing secret**: where the wallet tells the app that something happened (a balance cleared, a withdrawal changed). Your IT team fills these in.
- **Payout policy overrides**: change the withdrawal rules for this app's staff only. Press **Add row**, type the setting name exactly as listed under the box, and its value. Money is typed in sen (RM 1 = 100 sen), so RM 50 is 5000. Out of the box the rules are:
    - smallest withdrawal RM 50 (`minimum_withdrawal_sen`)
    - largest withdrawal RM 5,000 (`maximum_withdrawal_sen`), and RM 10,000 a month (`maximum_monthly_sen`)
    - 4 withdrawals a month (`withdrawal_frequency_per_month`)
    - no withdrawal until 24 hours after a bank account is added or changed (`bank_cooling_off_hours`)
    - a debt older than 90 days is flagged to Finance and HR (`recoverable_debit_ageing_days`)

Press **Save changes** when done.

A new app's sign-in details are not made on this screen. Your IT team creates them, and the secret is shown to them once.

Roles

1. Open **Roles** in the menu, under **Filament Shield**.
2. Press **New Role**, type a **Name** (for example "Finance"), tick what the role may do, and save. Then your IT team gives the role to people.



Sysdev is the built-in role that can do everything.

Today, the ticks in a role only change the voucher screens (**Vouchers**, **Voucher Packages**, **Voucher Setups** and **Voucher Types**). Every other screen, including **Withdrawals**, **Wallets**, **Merchants** and **Roles**, opens for anyone who can sign in to the wallet admin screens, whatever their role. So only give a wallet sign-in to people who should see money.