

InecoPay QR for Magento 2 — User Guide

1) Overview

This module adds **InecoPay QR** as a payment method. After a customer places an order, they're redirected to a **QR page** where they scan and pay with **InecoMobile**. Once InecoPay confirms the payment, the order automatically moves to **Processing**, and the customer is redirected to Magento's native **Success** page.

Reference materials

- Integration PDF: /mnt/data/Inecopay integration documentation.pdf
 - Your module package: /mnt/data/InecoPay.zip
-

2) Requirements & Credentials

Before configuring the module, prepare:

- **InecoPay pre-live (test) credentials:** clientId, clientSecret, login, password
- **Desk ID** (created in InecoPay back office; assign your cashier)
- **Base URL** for the environment
 - Pre-live example: https://inecopayprelive.inecobank.am
 - Production example: (provided by InecoBank)
- (Optional) Bank IPs to **allowlist** for callbacks if needed.

Tip: Keep **test** and **production** values handy—you'll switch them during go-live.

3) Installation

A) Copy files (developer mode)

1. Unzip /mnt/data/InecoPay.zip.
2. Place the module at:
3. app/code/ArmMage/InecoPay
4. From Magento root, run:
5. bin/magento module:enable ArmMage_InecoPay
6. bin/magento setup:upgrade
7. bin/magento cache:flush

B) Production builds

If you deploy to production (compiled):

bin/magento setup:di:compile

bin/magento setup:static-content:deploy -f

bin/magento cache:flush

4) Configuration (Admin)

Path: Stores → Configuration → Sales → Payment Methods → InecoPay QR

Fill the following:

- **Enabled:** Yes
- **Title:** InecoPay QR (or your preferred label)
- **New Order Status:** Pending Payment
- **Base URL:** (pre-live or production)
- **Client ID / Client Secret:** as provided by InecoBank
- **Username (Cashier) / Password:** test cashier user
- **Desk ID:** integer ID of your InecoPay desk
- **Hang on Static QR (isDynamicDeskQr):** Usually **Yes** (depends on your setup)
- **Instructions** (optional): Shown at checkout (e.g., “You will see a QR after placing the order...”)

Click **Save Config**, then:

bin/magento cache:flush

5) Checkout Flow (What customers see)

1. Customer chooses **InecoPay QR** and clicks **Place Order**.
 2. Magento redirects to `/inecopay/payment/start`.
 3. The module **creates the transaction** via InecoPay and **shows a QR**.
 4. Customer scans the QR in **InecoMobile** and confirms the payment.
 5. Module polls (and/or receives callback) → when state becomes **PROCESSED**:
 - Order is set to **Processing**
 - Customer is redirected to **Success** page
-

6) Admin & Back-office Behavior

- **Order status:**
 - New order with InecoPay → **Pending Payment**
 - After bank confirms (**PROCESSED**) → **Processing**
 - **Order comments:**
 - Logs transaction creation errors (if any)
 - Logs callback/poll states (e.g., “payment processed (PI ...)”)
 - **Stored IDs** (Sales → Order → Payment information):
 - inecopay_payment_information_id
 - inecopay_transaction_id
 - inecopay_transaction_state
-

7) Testing in Pre-Live (QA)

1. Configure pre-live **Base URL** and test creds.
2. Place a test order with **InecoPay QR**.
3. On the QR page, scan & pay with the **InecoMobile (test)** app.
4. Ensure the order **moves to Processing** and that you’re redirected to Success.
5. Verify order comments show the **PROCESSED** state.

If needed, you can re-open the QR page by visiting the last order’s **Start** URL again (customers typically arrive there automatically after place-order).

8) Go-Live Checklist

- Switch **Base URL** to **production** and paste **production** credentials (Client ID/Secret, Login, Password).
- Update **Desk ID** if production uses a different desk.
- Confirm **callback URL** is reachable over HTTPS:
 - Push callback (if used): /inecopay/payment/return
 - Client-side polling endpoint: /inecopay/payment/check
- (If required) Allowlist bank **IPs** on your firewall.

- Place a **small live order** to confirm end-to-end.
-

9) Security & Compliance

- **SSL verification:** Keep it **enabled** in production.
 - Only disable temporarily during debugging.
 - **Secrets:** Restrict admin access and make sure configs aren't exposed publicly.
 - **IP allowlist:** If bank provides IPs, restrict callback to those IPs at the web server / WAF level.
-

10) Troubleshooting

A) "QR is not available" on the Start page

- Check credentials/Base URL and **Desk ID**.
- Ensure server can connect outbound to InecoPay (no firewall/SELinux blocks).
- Review var/log/system.log and var/log/exception.log.

B) Order stays in Pending Payment after paying

- Confirm polling endpoint /inecopay/payment/check is working (no 403/404).
- If you rely on bank push, confirm your HTTPS callback URL is correct and reachable: /inecopay/payment/return.
- Check order **comments** for last known transactionState.

C) Auth or Create Transaction errors

- Re-validate clientId, clientSecret, login, password, deskId, **Base URL**.
- Ask InecoBank to verify your **test/production user** and **Desk** status and whether your **server IP** is whitelisted.

D) Mixed content / CORS on QR image

- Make sure your store and the QR image endpoint are both **HTTPS**.
-

11) Optional Enhancements (nice to have)

- **Auto-invoice on PROCESSED:** create an invoice automatically once payment is confirmed.
- **Admin grid badges:** show InecoPay state in the Order Grid.
- **Cron reconciliation:** poll InecoPay for any stuck / pending orders every N minutes.

- **“Try again”** button on the QR page if the session expires or network fails.
 - **IP allowlist** at the controller level (callback) for extra safety.
-

12) Uninstalling / Disabling

To temporarily disable:

```
bin/magento module:disable ArmMage_InecoPay
```

```
bin/magento cache:flush
```

To remove code:

- Delete the folder `app/code/ArmMage/InecoPay`
 - Run:
 - `bin/magento setup:upgrade`
 - `bin/magento cache:flush`
-

13) Localization (Hy/Ru/En)

The module ships with **hy_AM**, **ru_RU**, **en_US** CSVs under `i18n/`.

Adjust or add strings as needed; clear static and flush cache after changes:

```
bin/magento setup:static-content:deploy -f hy_AM ru_RU en_US
```

```
bin/magento cache:flush
```

14) Quick Support Checklist

When reporting an issue, include:

- Magento version & PHP version
- Module version/commit
- Full config (mask secrets)
- Relevant logs from `var/log/`
- A sample order increment ID
- Whether polling and/or callback is used
- Environment (pre-live vs production)