PayPal payments — add-on for the Code Craft HTML + PHP shop templates

Take payments with **PayPal** (PayPal balance, and cards through PayPal) using the **Orders v2** API. The order is captured when the customer comes back and marked *Paid* only when PayPal reports a completed capture for the right amount.

Product page: https://coodecraft.com/market/product/php-shop-paypal-addon

Works with the 12 Code Craft shop templates (site core 1.1 / shop engine 1.0 or later). It does not work with WordPress or with the non-shop templates.

How it works

1. The add-on gets an OAuth token (`/v1/oauth2/token`) and creates a PayPal order (`POST /v2/checkout/orders`, intent CAPTURE) for the exact total, then sends the customer to PayPal to approve it.
2. On the way back it checks the PayPal order belongs to this shop order and **captures** it (`POST /v2/checkout/orders/{id}/capture`). Only a capture with status COMPLETED and the right currency and amount marks the order paid.
3. A **webhook** (`PAYMENT.CAPTURE.COMPLETED`) is accepted only after PayPal's own **verify-webhook-signature** API answers SUCCESS for your webhook ID.
4. The shop engine never records the same capture twice.

Install

1. Upload the `paypal` folder into your website's `addons/` folder (so you have `addons/paypal/addon.php`).
2. Open **Admin** → **Add-ons**, switch on **PayPal payments** and open its **Settings**.
3. Paste your add-on licence key (from your purchase email; not needed on localhost) and your PayPal details (below).
4. In your PayPal app → Webhooks, add the URL shown on the add-on settings screen (`…/cc/endpoint.php?addon=paypal&action=webhook`) with the event **PAYMENT.CAPTURE.COMPLETED**, then paste the webhook ID.
5. Open **Admin** → **Shop** → **Payments**: the method is listed there. Keep it ticked and set its position.
6. Place a test order with sandbox credentials and a PayPal sandbox buyer account before you switch to live keys.

Settings

Setting	Meaning
Add-on licence key	Your key for this add-on, checked with coodecraft.com like the website's own licence (always active on localhost). Until it is active the method is not offered.
Sandbox mode	On for sandbox credentials, off for live.
Client ID / Secret	From developer.paypal.com → Apps & Credentials. The secret is never shown again.
Webhook ID	The ID PayPal shows next to the webhook you add.
Name at checkout	What customers see in the list of payment methods.

How it was tested

Real PHP 8.3 (`php -S`) on copies of the Kora and Pustak shop templates, with every PayPal API call answered by a local mock server (`config.php` → `api_base`), 47 checks per shop together with the other gateway add-ons: OAuth token, order created with the amount as a decimal string and the right currency, redirect to the payer-action link; a return with another PayPal order refused; a capture for a different amount → not paid (recorded as a mismatch); a completed capture → paid; a webhook PayPal does not verify → 400; a verified webhook → paid.

Limits and honest notes

- Tested only against **mocked** PayPal responses and locally forged callbacks/webhooks. It was **not** tested with real money or the live PayPal service. Try a test-mode payment yourself before going live.
- You need your own PayPal account. PayPal fees, payouts, refunds and disputes are handled in your PayPal dashboard; refunds are not sent from the website (set the order to *Refunded* yourself after refunding).
- The amount charged is the order total in your base currency, or in the customer's currency if you chose that in **Shop settings** (at your rate).
- PayPal only supports some currencies; INR works only for Indian PayPal accounts paid in India. Choose your base currency (or "charge in the customer's currency") accordingly.

Files

- `addon.php`: the manifest (settings shown in Admin → Add-ons).
- `paypal.php`: the code (payment method, return/callback and webhook endpoints).
- Details of the add-on contract: `cc/ADDONS.md` on your website.

The settings screen

Admin → Add-ons → paypal, on a local copy of the Kora shop template:

