

Phoenix One Page Checkout

V1.0.2

CE Phoenix Cart 1.1.0.6 / 1.1.0.7 · V1.0.2 · 30 September 2026

One Page Checkout is a no-core-change checkout replacement for CE Phoenix Cart 1.1.0.6 and 1.1.0.7. Customer details, addresses, delivery, discounts, payment selection, payment confirmation and order review remain on one store page. A hosted payment provider may still take the customer to its own secure website.

OPC asks Phoenix which payment modules are installed, enabled and eligible for the current order. It runs the normal selection, JavaScript validation, pre-confirmation, confirmation and process-button methods, while retaining the standard `checkout_confirmation` form contract. Ordinary offline methods can therefore work without being added to a whitelist. Hosted, iframe and browser-SDK methods must still be tested because their page, callback and JavaScript assumptions differ.

Use Test/Sandbox accounts and complete the applicable cases in [docs/TEST_PLAN.md](#) before live use.

V1.0.2 highlights

- One package supports CE Phoenix Cart 1.1.0.6 and 1.1.0.7, including the newer PHP 8.4 form/session behaviour.
- Payment modules are discovered from Phoenix rather than a fixed OPC whitelist.
- `payment->selection()` runs only while the payment choices are displayed, avoiding duplicate heavy SDK initialisation during confirmation.
- The standard `checkout_confirmation` form name, `inputComments` ID and `btn-success` confirmation class are retained for payment-module compatibility.
- Payment confirmation, embedded fields and the final button remain on `one_page_checkout.php`; the legacy handoff URL is only a compatibility redirect.
- A sole valid delivery method is selected automatically and its configured service name is shown without the technical `Multi Table Shipping` prefix.
- Automatically supports either optional Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 or Raiwa PWA Guest Checkout 5.4; guest wording and controls disappear when neither is available.
- Signed-in customers can select, add and edit delivery and billing addresses.
- Selected product attributes are shown beneath each product in the order summary.
- The native Phoenix cart Checkout button is used when available; OPC supplies one fallback button only when it is not.
- Stripe Payment Element receives a validated, one-use return path for its JavaScript-led payment flow.

Payment compatibility status

Automatic discovery means a method can appear; it does not certify that every callback, iframe or page-name assumption is compatible. These are the V1.0.2 results:

Tested with OPC	Not yet fully tested with OPC
Stripe Hosted Payments V1.0 on Phoenix 1.1.0.6	Phoenix-supplied PayPal Standard complete Sandbox flow on both supported cores

Tested with OPC	Not yet fully tested with OPC
Cartmart Stripe Payment Element 1.2.12+ , developer-confirmed with the OPC submit/return contract	Phoenix Stripe Payment / Stripe SCA complete card, decline and 3-D Secure flow
Bank Transfer on Phoenix 1.1.0.6	Stripe Hosted Payments on Phoenix 1.1.0.7
Phone with Card Payment on Phoenix 1.1.0.6	Every other automatically discovered payment module until its full flow is recorded

Cartmart Stripe Payment Element also has a [Phoenix add-ons listing](#). Version 1.2.12 or later is required because that release ensures OPC's submit validation is invoked before the Stripe JavaScript continues.

PayPal Standard is already supplied with Phoenix: version 1.5 in Phoenix 1.1.0.6 and version 1.6 in Phoenix 1.1.0.7. Do not replace those bundled versions with an older download from the add-ons listing. V1.0.2 does not contain or replace any PayPal payment-module file.

The optional Stripe folder is only for the separately installed Stripe Hosted Payments V1.0 add-on. Stripe Payment Element, Stripe SCA, PayPal Standard, Bank Transfer and Phone with Card Payment are not bundled with OPC.

Acknowledgements - John Ferguson

Particular thanks and substantial credit are due to John Ferguson (@BrockleyJohn) of [Cartmart](#). John independently installed and tested the checkout across Phoenix 1.1.0.6 and 1.1.0.7, identified the 1.1.0.7/PHP 8.4 form-session difference, traced duplicate payment-selection execution and the resulting PayPal SDK/`zoid destroyed all components` failure, and supplied the `paymentSelectionPreload` hook introduced with PPCHECKOUT 1.4.6. He also reworked Stripe Payment Element 1.2.12 so OPC's submit validation is called before its browser payment flow, then verified the completed integration.

That practical testing and developer feedback materially shaped the dual-core session handling, stage-aware SDK preload suppression, main-page confirmation flow and Stripe Payment Element return protection in this release. The project is markedly safer and more compatible because of his contribution.

Guest-integration credit remains with Mort-Lemur for Phoenix Guest Checkout and Rainer Schmied (@raiwa) for PWA Guest Checkout 5.4. Neither original guest add-on is bundled; [optional_guest_integrations/README.md](#) defines the exact integration contract.

Scope

- Physical-product baskets; virtual, download and mixed baskets use standard checkout.
- Guest purchasing optionally uses either Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 or Raiwa PWA Guest Checkout 5.4. Signed-in checkout works without either add-on.
- The separately labelled Mort-Lemur companion is only for Phoenix Guest Checkout 1.0.0-RC3. Raiwa PWA requires no replaced file.
- Optional Discount Codes 6.3 support.
- Hosted cancellation or recoverable failure returns to the populated checkout when supported by the payment module or adapter.
- English language files and the default Phoenix template are included.

No raw Stripe, PayPal or bank-login credentials are collected by OPC. Stripe Elements send card data directly to Stripe. Phone with Card Payment instructs the customer to telephone the store and never to submit card details online.

The printable manuals are in [docs/](#).

Licence

Copyright (c) 2026 **Mort-Lemur**. Released under the GNU General Public License; see [LICENSE.txt](#).

Installation

Requirements

- CE Phoenix Cart 1.1.0.6 or 1.1.0.7.
- HTTPS throughout checkout.
- At least one installed, enabled and order-eligible Phoenix payment module.
- Optional Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 or Raiwa PWA Guest Checkout 5.4 only when guest purchasing is required. Neither is required for signed-in checkout.
- Optional Discount Codes 6.3 when discount entry is required.
- A complete Test/Sandbox run for every enabled payment method before live use.

The main OPC **catalog** contains no payment-module or guest-add-on replacements. Bank Transfer, Phone with Card Payment, Stripe Payment Element, Stripe SCA, PayPal Standard and both supported guest add-ons remain separate packages.

Back up files and the database before installing.

Installation

1. Merge this package's **catalog** directory into the store catalog root.
2. In Admin > Modules > Content > Shopping Cart, install One Page Checkout and set it to **True**.
3. Keep the normal Phoenix Checkout shopping-cart content module installed and enabled where possible. OPC redirects its destination. If it is missing or disabled, OPC supplies one fallback Secure Checkout button automatically.
4. Confirm the required payment modules are installed, enabled and configured.
5. Use payment Test/Sandbox modes until every relevant test in **TEST_PLAN.md** passes.

The OPC content module normally acts as a controller only. It renders a fallback button only when Phoenix's native Checkout module is unavailable, and needs no width or sort-order setting.

No Phoenix core file is replaced.

Optional folders

Upload an optional folder only when the exact named add-on/version is already installed:

- **optional_payment_integrations/stripe_checkout_v1_0/catalog** - [Stripe Hosted Payments V1.0](#) only.
- **optional_guest_integrations/mort_lemur_guest_checkout_1_0_0_rc3/catalog** - Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 only.

Do not upload these folders to stores without the matching add-on. Back up every replaced add-on file first.

Phoenix's supplied PayPal Standard and Stripe payment module do not use an optional folder. Cartmart Stripe Payment Element is separately installed and is not included here.

Optional Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3

Install the original add-on first, then merge

`optional_guest_integrations/mort_lemur_guest_checkout_1_0_0_rc3/catalog`. The companion changes only `includes/hooks/shop/siteWide/pgcGuestCheckout.php`. It keeps `customer_id` available until success-page outgoing modules finish, then logs out the guest and lets Continue Shopping reach the storefront. It checks that the `pgc` class and required methods exist before doing any work, so a missing or incompatible add-on cannot produce the earlier class-not-found fatal.

Optional Raiwa PWA Guest Checkout 5.4

Install Raiwa's original package normally; OPC contains no Raiwa replacement files. Enable Login without Account Form and the Guest customer-data module. Assign `checkout_guest` to Email and all customer-data modules required by the store's address, delivery, payment and legal rules. Keep Redirect to Create Account PWA set to `False`, because OPC presents guest checkout directly beside Sign In. Install PWA Keep Account only if account conversion is wanted after purchase.

OPC embeds Raiwa's configured `checkout_guest` fields, sets its native `customer_is_guest` session marker and leaves its order email, private order link, optional account conversion and success cleanup under Raiwa's control.

Install only one guest-checkout add-on. If neither supported add-on is installed and available, OPC shows no guest heading, warning or form.

Payment-module notes

- [Cartmart Stripe Payment Element](#) must be version 1.2.12 or later. OPC validates the final form before its JavaScript calls Stripe and permits a one-use GET return to `checkout_process.php?action=process`. The payment module remains responsible for verifying the returned `PaymentIntent` before finalising the order.
- Phoenix PayPal Standard is supplied with the core: 1.5 on Phoenix 1.1.0.6 and 1.6 on Phoenix 1.1.0.7. Do not downgrade it using an older add-ons download.
- Phoenix [Stripe Payment / Stripe SCA](#) keeps its on-site Stripe.js Elements and 3-D Secure lifecycle.
- Configure [Bank Transfer](#) as Pending when that is the required business process.
- Configure [Phone with Card Payment](#) with an unpaid status, telephone number, hours and instructions.
- Stripe Hosted Payments intentionally remains off site.

The main OPC catalog includes `includes/hooks/shop/one_page_checkout/opcHooks.php`. For a payment module that exposes `paymentSelectionPreload`, this page-scoped hook leaves the useful SDK preload enabled when choices are first displayed and suppresses it only after OPC advances into confirmation. John Ferguson (@BrockleyJohn) supplied the original preload hook; OPC adds the stage-aware condition.

Automatic discovery means ordinary offline modules normally need no whitelist entry. Hosted, callback-based, iframe or browser-SDK modules still require a complete compatibility test and may require a small OPC adapter.

First checks

1. Confirm there is exactly one Checkout button in the basket.
2. Confirm the checkout session survives every address, delivery and payment selection on the store's Phoenix version.
3. Without either supported guest add-on installed, confirm only Sign In and Create Account are shown.
4. With Mort-Lemur Guest Checkout installed, confirm its guest form, private order link and post-success logout.

5. In a separate installation with Raiwa PWA 5.4, confirm its guest form, email/private order link and configured delete-or-convert success choice.
6. Confirm a sole delivery method is selected automatically and clearly labelled.
7. Confirm payment selection fields and the selected module's confirmation output appear on the main OPC page.
8. Complete, cancel and fail every enabled payment method in Test/Sandbox mode.
9. Confirm order email, status, stock, totals, basket clearing and duplicate-order protection.
10. For a module with an SDK preload hook, confirm the SDK is preloaded on the choice display but not loaded a second time during confirmation.
11. For Stripe Payment Element, test the paid GET return and rejected direct/replayed GET requests.

Uninstall

1. Remove One Page Checkout in Admin.
2. Delete the main OPC files listed in **MANIFEST.txt**.
3. Restore any add-on files replaced from the optional Stripe integration folder.
4. Restore the original Mort-Lemur Guest Checkout hook if its RC3 companion was applied. OPC does not replace a Raiwa PWA file.
5. Leave or re-enable Phoenix's normal Checkout shopping-cart module.

Removing OPC does not remove orders or payment add-on audit data.

Compatibility

Source targets

Component	V1.0.2 target
CE Phoenix Cart	1.1.0.6 and 1.1.0.7
PHP	Phoenix-supported PHP 8.2-8.4; test on the store's exact version
Template	Phoenix default template
Basket	Physical products
Guest checkout	Optional Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 or optional Raiwa PWA Guest Checkout 5.4
Phoenix PayPal Standard	Bundled 1.5 on Phoenix 1.1.0.6; bundled 1.6 on Phoenix 1.1.0.7
Phoenix Stripe Payment / SCA	Current core/add-on version appropriate to the Phoenix release
Cartmart Stripe Payment Element	Version 1.2.12 or later, installed separately
Optional Stripe add-on	Stripe Hosted Payments V1.0 folder
Offline examples	Bank Transfer 1.0.0.1; Phone with Card Payment V1.0
Discounts	Discount Codes 6.3 where installed

Source target means the integration has been designed and statically reviewed against those versions. The store must still complete the relevant guest and payment Test/Sandbox cases.

Tested and pending payment coverage

Tested with OPC	Not yet fully tested with OPC
Stripe Hosted Payments V1.0 - successful hosted payment on Phoenix 1.1.0.6	Phoenix-supplied PayPal Standard - complete Sandbox success/cancel/failure run on both supported cores
Cartmart Stripe Payment Element 1.2.12+ - completion confirmed after its submit chain was made OPC-aware	Phoenix Stripe Payment / Stripe SCA - complete card, decline and 3-D Secure run
Bank Transfer - Pending status, stock, email and basket clearing on Phoenix 1.1.0.6	Stripe Hosted Payments V1.0 on Phoenix 1.1.0.7
Phone with Card Payment - local order flow on Phoenix 1.1.0.6	Any other auto-discovered payment module until its success, cancellation and failure paths are recorded

“Tested” records the completed integration evidence available for V1.0.2; it is not a substitute for testing the merchant's own credentials, PHP version, theme, hooks and extensions. Stripe Payment Element also has a [Phoenix add-ons listing](#).

Optional guest-checkout systems

Install only one guest system. Neither original add-on is bundled with OPC.

Guest system	Detection and responsibility
Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3	OPC detects the complete <code>pgc</code> interface and uses its guest field, reusable-customer and session methods. The separately labelled companion hook supplies the tested success/logout ordering. Mort-Lemur's add-on remains responsible for private tokens, order access, email and conversion.
Raiwa PWA Guest Checkout 5.4	OPC requires its enabled Login without Account and Guest modules, the selected template's <code>pwa.php</code> hook, and <code>email_address</code> plus <code>guest</code> on <code>checkout_guest</code> . OPC uses those fields and starts <code>customer_is_guest</code> ; Raiwa's hook remains responsible for order email, private order access, optional conversion and success cleanup. No Raiwa file is replaced.
Neither available	OPC exposes no guest heading, warning or form and continues to support registered customers.

Raiwa PWA 5.4 states compatibility with Phoenix 1.1.0.0 onward and was tested by its author with Phoenix 1.1.0.7. The V1.0.2 adapter has been statically reviewed against the supplied 5.4 package; its complete runtime matrix remains part of [TEST_PLAN.md](#).

Customer presentation

Store/customer state	Behaviour
Both supported guest add-ons absent or unavailable	No guest heading, warning or form is shown; only Sign In and Create Account are offered.
One supported guest add-on available	That add-on's configured guest fields are shown beside the returning-customer route.
Signed-in customer	Saved addresses can be selected, edited or supplemented.
Signed-out account customer	Phoenix login returns the customer to OPC through the navigation snapshot.

Payment contract

V1.0.2 asks Phoenix for installed modules, constructs the current order and calls `payment->selection()` once while displaying the payment choices. It retains Phoenix's standard confirmation contract:

- Form name: `checkout_confirmation`
- Form ID before a module changes it: `check_form`
- Comments field ID: `inputComments`
- Final button includes: `btn-success`
- Methods called: selection, JavaScript validation, pre-confirmation, confirmation and process button

Payment modules with no selection fields can be selected immediately. If a selection contains fields or iframe UI, OPC does not automatically submit the payment. The customer or module completes its fields and confirms readiness. The selected module's confirmation fields and scripts retain control of final submission.

For modules implementing `paymentSelectionPreload`, the page-specific OPC hook is stage-aware. It allows preload when payment choices are first displayed and suppresses it only when a valid choice advances into confirmation or that confirmation is already prepared. This prevents duplicate SDK/component initialisation while retaining the intended connection warm-up.

John Ferguson ([@BrockleyJohn](#)) supplied the original payment-selection preload hook and diagnosed the duplicate SDK behaviour. Mort-Lemur added the OPC-stage condition.

Method	OPC behaviour
Ordinary offline/local module	Revalidates token, customer, cart, address, delivery, payment, totals, terms and stock before Phoenix <code>checkout_process.php</code> .
Phoenix Stripe Payment / SCA	Places its Stripe.js Elements on the main OPC confirmation form; its JavaScript controls submission and 3-D Secure. Full V1.0.2 runtime coverage remains pending.
Phoenix PayPal Standard	Prepares the cross-site session cookie, validates the form before departure, then leaves IPN/PDT and return verification to the bundled module. Full V1.0.2 Sandbox coverage remains pending.
Cartmart Stripe Payment Element 1.2.12+	Validates form state before JavaScript confirmation and permits a nonce-bound, one-use GET return to <code>checkout_process.php?action=process</code> . The module verifies the Stripe result before finalisation.
Module with <code>paymentSelectionPreload</code>	Allows initial choice-display preload and suppresses it during confirmation.
Stripe Hosted Payments V1.0	Uses the optional adapter and intentionally redirects to Stripe's hosted page.
Unknown hosted/iframe module	Discovered when Phoenix reports it eligible, but must be tested; page-name, form, DOM or callback assumptions may require an adapter.

Phoenix 1.1.0.6 versus 1.1.0.7

Phoenix 1.1.0.7 changed form/session handling for PHP 8.4 and no longer relies on the deprecated SID constant in the same way. V1.0.2 does not hard-code either implementation. Every internal OPC form calls the active Phoenix version's `Form::hide_session_id()`, and internal links no longer suppress Phoenix's session parameter.

PayPal Standard changes from 1.5 to 1.6 between these core versions. V1.0.2 replaces neither version. The add-ons listing may contain an older downloadable release; do not use it to downgrade the version supplied with the core.

Limits

- Virtual, download and mixed baskets use standard checkout.

- A module can be automatically discovered yet still be incompatible because it hard-codes native page names, DOM IDs, callback routes or an unusual order-creation sequence.
- Zero-balance orders are not a V1.0.2 OPC payment path.
- Non-default templates need an adapted page template.
- Additional languages need translated language files.

Security Notes

- OPC does not collect raw hosted-provider credentials. Stripe SCA's Elements send card data directly to Stripe. Phone with Card Payment explicitly keeps card details off the website, comments and email.
- Every checkout-changing internal POST uses Phoenix's session form token. Internal forms call the active Phoenix version's `Form::hide_session_id()` for 1.1.0.6/1.1.0.7 compatibility.
- Address selection/editing verifies that the address belongs to the current customer and then uses Phoenix customer-data validation.
- Delivery quotes are rebuilt and matched to the submitted method.
- Payment codes are restricted to safe class-name characters, must be present in `MODULE_PAYMENT_INSTALLED`, and must be returned as available by Phoenix for the current order.
- Payment selection is executed once during display. Confirmation uses only the selected installed/enabled module.
- Local finalisation revalidates the form token, customer, cart ID, address/delivery state, payment module, order fingerprint, terms and stock before issuing and consuming a nonce-bound one-use permission for `checkout_process.php`.
- External/custom form actions first call `one_page_checkout_validate.php` with the same token and current form data. Only a successful same-origin validation arms the one-use return permission; the payment module then retains control of its own submission and callback verification.
- The Stripe Payment Element form receives the same validation before its JavaScript confirms payment. Its GET return is accepted only for the selected `stripe_element` module when a nonce-bound permission exists and `action=process` is present. The permission is consumed once; a bare GET without prior confirmation-form validation is refused. The permission does not prove that payment succeeded: after preflight and before paying, a direct GET remains possible. The vendor module must independently verify the PaymentIntent before creating or marking an order as paid.
- External payment forms do not receive a hidden session ID. PayPal Standard instead receives the same `Secure; SameSite=None` session-cookie policy used by native confirmation.
- Provider modules retain responsibility for amount, currency, merchant, ownership, signature, idempotency, capture/session and webhook verification.
- Cancellation/failure does not clear the basket. Checkout-success clears OPC state.
- Guest access, private order tokens and account conversion remain the responsibility of the selected guest add-on. Mort-Lemur Guest Checkout uses its `pgc` interface and optional RC3 companion; Raiwa PWA 5.4 uses its native `customer_is_guest`, order-mail and success-page hooks. OPC does not accept a partial provider contract and exposes no guest form when neither supported provider is complete.
- Before Raiwa creates a new guest for an email, OPC mirrors Raiwa's native stale-guest cleanup. It selects only a customer already marked `customers_guest = 1`, removes that guest's customer/address/basket rows and leaves historic orders intact. An existing normal account is never deleted or silently converted.
- Install only one supported guest add-on. No guest add-on is bundled with OPC, and the Raiwa integration replaces no Raiwa file.

- The legacy handoff URL performs no payment work and immediately redirects to the main checkout.

Reporting

Use test/Sandbox accounts. Do not share credentials, live payment data, guest tokens or customer personal information. Redact logs before sending them to a developer.

V1.0.2 Test Plan

Use Stripe Test mode and PayPal Sandbox. Start each numbered group with a fresh basket unless the case explicitly tests retry behaviour. Record the Phoenix order ID and provider reference for each payment attempt.

1. Entry and fallback

- ☐ A physical basket opens One Page Checkout from the single native Phoenix Checkout button.
- ☐ No second OPC checkout button is rendered in the basket.
- ☐ With Phoenix's native Checkout content module disabled, OPC supplies exactly one Secure Checkout button.
- ☐ MZMT shows the configured service name, for example Courier / Haulier, without the technical **Multi Table Shipping** prefix.
- ☐ On the installed Phoenix version, the session, customer and cart survive every OPC form selection.
- ☐ Direct access with an empty basket returns to the shopping cart.
- ☐ Virtual, downloadable and mixed baskets enter standard Phoenix checkout.
- ☐ Unsupported basket types fall back without presenting a customer-facing standard-checkout link.
- ☐ Returning to OPC re-enables its replacement flow after an automatic fallback.

2. Guest and account customers

- ☐ With both supported guest add-ons absent or disabled, no guest heading, unavailable warning or guest form is shown; Sign In and Create Account remain available.
- ☐ Test Mort-Lemur Phoenix Guest Checkout 1.0.0-RC3 and Raiwa PWA Guest Checkout 5.4 in separate installations; do not enable both together.
- ☐ For each provider, valid details create its native guest session and invalid details remain on the page with validation.
- ☐ For each provider, an email belonging to a normal account is not deleted or silently converted to a guest login.
- ☐ Mort-Lemur's successful guest email contains its working private order link; success-page outgoing modules receive **customer_id**, the guest is then logged out and Continue Shopping reaches the storefront.
- ☐ Raiwa's successful guest email contains its working private order link; the initial success page retains **customer_id**, then its configured delete-or-convert choice completes without warnings or orphaned session data.
- ☐ Raiwa's existing abandoned guest record can be safely reused by email without deleting historic orders, while a normal account with the same email remains protected.
- ☐ A signed-in customer sees the correct name, email and owned addresses.
- ☐ Delivery and billing addresses can be selected, edited and added independently.
- ☐ A tampered address ID belonging to another customer is rejected.
- ☐ A signed-out account customer signs in and returns to OPC.

3. Shipping, totals and discounts

- ☐ Each selected product attribute is shown beneath the correct product in the order summary, including products with more than one attribute.
- ☐ Product names and attribute names/values containing punctuation or HTML-sensitive characters display as text without breaking the summary layout.
- ☐ Quotes match standard checkout for the same address and basket.
- ☐ When exactly one valid delivery method exists, it is selected automatically and payment becomes available without a click.
- ☐ When two or more delivery methods exist, none is selected automatically.
- ☐ A shipping error is shown without enabling payment.
- ☐ Changing delivery address or shipping invalidates the old payment choice.
- ☐ Free shipping can be selected when eligible.
- ☐ Valid Discount Codes 6.3 changes totals correctly; invalid, expired and ineligible codes are rejected.
- ☐ Removing a code restores totals.
- ☐ Tax, delivery, discount and grand total match standard checkout.

4. Terms, stock and tampering

- ☐ Required terms/privacy acceptance blocks continuation when unticked and shows the error beside the legal agreement rather than inside a provider panel.
- ☐ Comments survive local validation, provider cancellation and recoverable errors.
- ☐ An unavailable payment ID submitted manually is rejected.
- ☐ A stale or tampered shipping ID is rejected.
- ☐ Reducing stock before final confirmation blocks hand-off when Phoenix disallows overselling.
- ☐ Changing the basket in another tab makes stale confirmation data fail safely.
- ☐ A checkout prepared more than two hours earlier returns safely to OPC.

5. Generic payment-module contract

- ☐ Every installed, enabled and eligible method appears without adding its code to an OPC whitelist.
- ☐ Uninstalled, disabled, zone-ineligible or errored methods do not appear.
- ☐ `payment->selection()` runs only while payment choices are displayed and does not initialise an SDK twice during confirmation.
- ☐ Selection fields and module JavaScript initialise correctly.
- ☐ Pre-confirmation, confirmation output and process-button methods run in the standard order.
- ☐ The form is named `checkout_confirmation`, comments use `inputComments`, and the final button includes `btn-success` unless the module intentionally changes them.
- ☐ Confirmation output and the final button remain on `one_page_checkout.php`; the legacy handoff URL only redirects.
- ☐ A method with fields or iframe UI is not auto-submitted; the customer or module confirms readiness.
- ☐ A module with `paymentSelectionPreload` may preload during the choice stage but is suppressed during prepared confirmation.
- ☐ The browser console contains no duplicate-SDK or `zoid destroyed all components` error.

6. Stripe Hosted Payments V1.0

- ☐ Test payment redirects to hosted Stripe Checkout.
- ☐ Cancel returns to populated OPC with a cancellation message.
- ☐ Successful card payment reaches Phoenix checkout success.
- ☐ Declined card or recoverable failure retains the basket and permits another attempt.
- ☐ Refresh, Back and webhook/browser-return ordering cannot finalise twice.
- ☐ Amount, currency, customer, cart and session references match the Phoenix order.
- ☐ Creation, webhook and return produce no duplicate Stripe language-constant warnings.
- ☐ Repeat the complete group on Phoenix 1.1.0.7 before marking that core tested.

7. Cartmart Stripe Payment Element 1.2.12+

- ☐ Confirm the installed version is 1.2.12 or later and Phoenix reports `stripe_element` enabled and eligible.
- ☐ Payment Element fields enable the final button only when ready.
- ☐ Required terms acceptance is validated before the Stripe SDK starts.
- ☐ OPC's validation endpoint succeeds once for a valid final form; the Stripe SDK then confirms payment and returns by GET to `checkout_process.php?action=process` with the same session.
- ☐ The module verifies the PaymentIntent and creates exactly one order with the correct status, stock and email.
- ☐ A direct GET before a valid final-form preflight is rejected without creating an order.
- ☐ A direct GET after preflight but before a successful Stripe payment cannot create or mark an order paid.
- ☐ A replayed GET, refresh or Back/Forward navigation cannot create a duplicate order or email.
- ☐ Missing Terms, invalid form, declined card and abandoned authorisation do not create a paid order.
- ☐ Cancellation and asynchronous methods follow the module's verified status and webhook rules.
- ☐ Repeat the successful, declined and 3-D Secure cases on each supported Phoenix/PHP combination used by the store.

8. Phoenix Stripe Payment / Stripe SCA

- ☐ The module appears only when installed, enabled, correctly configured and eligible.
- ☐ Selecting it mounts secure Stripe.js Elements on the main OPC page without a second checkout page.
- ☐ The module can alter the form ID as expected and retains one usable confirmation button.
- ☐ Invalid and declined cards remain on OPC with a useful error and allow another attempt.
- ☐ A successful card and a 3-D Secure test finalise once and reach checkout success.
- ☐ Retry, refresh and Back navigation do not duplicate a prepared/final order.
- ☐ Stock, totals, status, email, basket clearing and guest logout are correct.
- ☐ Raw card data never reaches Phoenix or OPC.

9. Phoenix-supplied PayPal Standard

- ☐ Use the PayPal Standard version supplied with the core: 1.5 on Phoenix 1.1.0.6 or 1.6 on Phoenix 1.1.0.7.
- ☐ It appears only when installed, enabled, configured and eligible for the billing zone.

- ☐ OPC refreshes the session cookie with **Secure** and **SameSite=None** before posting to PayPal Sandbox.
- ☐ The module creates only its normal prepare-status order and posts the expected amount, currency, invoice and customer reference.
- ☐ A comment is copied once to the prepare/final order history and email.
- ☐ Cancelling at PayPal returns to populated OPC with the cancellation message and retained comments/selections.
- ☐ Successful Sandbox payment resumes the same customer/cart session and reaches checkout success.
- ☐ PDT, IPN and browser return cannot finalise the order twice.
- ☐ A rejected or unverifiable return fails through PayPal Standard without OPC bypassing its verification.
- ☐ Stock, totals, final status, history, email, basket clearing and guest logout are correct.

10. Bank Transfer

- ☐ **banktransfer** appears only when installed, enabled and valid for the payment zone.
- ☐ OPC displays the configured receiving-account details.
- ☐ Confirming creates one local order without redirecting to a provider.
- ☐ The module is configured and verified as Pending.
- ☐ The order email contains the bank details.
- ☐ Refresh and Back do not create a duplicate order.

11. Phone with Card Payment

- ☐ **phonecard** appears only when installed, enabled and valid for the payment zone.
- ☐ OPC displays the telephone number, opening hours, instructions and card-data security warning.
- ☐ Confirming creates one local order without a second checkout page or provider redirect.
- ☐ The configured unpaid status is used.
- ☐ The order email contains the telephone-payment instructions and order-number reminder.
- ☐ No checkout field accepts card details, and refresh or Back does not duplicate the order.

12. Order finalisation

- ☐ The basket is retained on cancellation/failure and cleared only after verified finalisation.
- ☐ Stock and discount usage update once.
- ☐ Order history/status is correct for paid, review and pending outcomes.
- ☐ Customer/admin notifications send once with correct totals, comments and payment information.
- ☐ A guest token is issued once and remains private.
- ☐ **checkout_success.php** clears OPC state.

Exit criteria

There must be no open critical or high defects. Every enabled payment method and selected guest provider must pass its success, cancellation/failure, idempotency, ownership, totals, stock and guest/session cases on the merchant's actual Phoenix/PHP configuration. Repeat the relevant group after any payment-module, guest-add-on, Phoenix, PHP or theme update.