WooCommerce Cheat Sheet

WooCommerce 11.2.1 (2026-10-09) · **WordPress 7.1.3** · PHP 8.3 · WP-CLI · Action Scheduler 4.1.0
Next: 11.3 (beta 1 2026-10-26) · Checked 2026-10-27

Versions, cadence, requirements

- New minor **about every 5 weeks**: freeze (day 1, `release/x.y`), beta 1 +1 wk, beta 2 +1 wk, RC 1 +1 wk, final +1 day. `x.y.z` is not semver; `z` > 0 = patch.
- Recent: 11.0.0 2026-08-04, 11.1.0 2026-09-03, 11.2.0 2026-10-07, 11.2.1 2026-10-09; 11.3 planned week of 2026-11-09.
- **L-1 policy**: latest + previous WordPress and WooCommerce series; latest patch only.
- **Security backports** (CVSS ≥ 9): last **21** `x.y` series = **9.2+** today; below CVSS 9 only the latest.

Source	WP	PHP	DB
Header 11.2.1	7.0	7.4	-
readme.txt	7.0	7.4 (8.0 rec.)	MySQL 5.5.5 / MariaDB 10.1
Merchant page (10.8+)	6.9	8.3+ (7.4 legacy)	MySQL 8.0 / MariaDB 10.6

11.1.0 raised WP minimum 6.9 → 7.0. **PHP 8.1 is the minimum from 11.6 (Feb 2027)**; 11.3 adds an admin notice for PHP 7.4/8.0. The `WC requires at least` and `WC tested up to` header keys are a WooCommerce convention; WordPress ignores them.

Install and local environments

```
wp core download --version=7.1.3
wp config create --dbname=... --dbuser=...
wp core install --url=... --title=...
wp plugin install woocommerce --version=11.2.1 --activate
wp wc hpos enable --user=admin  # off after WP-CLI install
```

- Docker: `wordpress:7.1.3-php8.3-apache`, `mariadb:11.8`, `wordpress:cli-php8.3`.
- **wp-env** 12.0.0: `npx @wordpress/env@12.0.0 start`; `.wp-env.json` keys `core`, `phpVersion`, `port` (8888), `plugins`, `config`, `testsEnvironment`. Use `run cli -- wp ...` (the `--` matters). Change versions in the JSON, then `start --update`.
- **Playground CLI** `@wp-playground/cli` 3.1.61 (Node 20.18+, no Docker); `wp-now` is deprecated.

Extension skeleton

```
/* Plugin Name: My Extension  Version: 1.0.0
 * Requires at least: 7.0  Requires PHP: 7.4
 * Requires Plugins: woocommerce
 * WC requires at least: 11.1  WC tested up to: 11.2 */
defined( 'ABSPATH' ) || exit;
add_action( 'before_woocommerce_init', static function () {
 if ( class_exists( FeaturesUtil::class ) ) {
  FeaturesUtil::declare_compatibility(
   'custom_order_tables', __FILE__, true );
  FeaturesUtil::declare_compatibility(
   'cart_checkout_blocks', __FILE__, true );
 } } );
add_action( 'plugins_loaded', 'x_bootstrap' );
// bootstrap: class_exists('WooCommerce') or notice
```

- `declare_compatibility()` works **only** inside `before_woocommerce_init` (false elsewhere); third arg `false` = incompatible. Check: `wp wc hpos compatibility-info`.
- Scaffold: `npx @wordpress/create-block -t @woocommerce/create-woo-extension my-ext`; add `uninstall.php` and `languages/`.

Hooks: classic vs blocks checkout

Data hooks fire everywhere: `woocommerce_new_order`, `woocommerce_update_order`, `woocommerce_order_status_changed`, `woocommerce_cart_calculate_fees`, `woocommerce_check_cart_items`. Markup hooks fire only on shortcode pages.

Classic(notfiredinblocks)	Block replacement
`woocommerce_checkout_fields`	`woocommerce_register_additional_checkout_field()`
`woocommerce_after_checkout_validation`	`woocommerce_checkout_validate_order_before_payment`, `woocommerce_store_api_cart_errors`
`woocommerce_checkout_create_order`, `_update_order_meta`	`woocommerce_store_api_checkout_update_order_from_request`
markup hooks, `..._cart_item_name`	Slot and Fill, inner blocks, JS `registerCheckoutFilters()`

A hook absent from the official hook-alternatives list is "unknown": trace it on your own store.

CRUD objects and HPOS rules

```
$order = wc_get_order( $id );
$order->update_meta_data( '_pc_note', $v ); $order->save();
$ids = wc_get_orders( array( 'limit' => 20, 'status' => 'processing', 'return' => 'ids' ) );
$p = new WC_Product_Simple(); $p->set_name('x');
$p->set_regular_price('9.50'); $p->save();
wc_get_products( array( 'sku' => 'PC-MUG' ) );
```

- **Never** use `get_post`, `WP_Query`, `get_post_meta`, `update_post_meta`, `wp_insert_post` for orders; use CRUD + `meta_query`/`field_query`/`date_query` in `wc_get_orders()`.
- HPOS tables: `wc_orders`, `wc_order_addresses`, `wc_order_operational_data`, `wc_orders_meta`; with sync off posts hold only `shop_order_placehold`. Default for new shops since **8.2**; sync-on-read off since 10.7.0.
- `wp wc hpos` status | enable [--with-sync] | disable | sync | verify_data | compatibility-info.
- **Action Scheduler**: `as_enqueue_async_action($hook,$args,$group,$unique)`, `as_schedule_single_action($ts,$hook,...)`, `as_schedule_recurring_action($ts,$secs,...)`; CLI `wp action-scheduler run|clean|status`.

Additional Checkout Fields (Checkout block)

```
add_action( 'woocommerce_init', function () {
 woocommerce_register_additional_checkout_field( array(
  'id' => 'pocket-coffee/gift-message',
  'label' => __( 'Gift message', 'td' ),
  'location' => 'order', // contact|address|order
  'type' => 'text',
  'attributes' => array( 'maxLength' => 120 ),
  'sanitize_callback' => fn( $v ) => trim( $v ),
  'validate_callback' => fn( $v ) => ... ) ); } );
```

- Id is `namespace/name`; types `text`, `checkbox`, `select` (`options`) and `date` (`min`/`max` like `P1D`, new in 11.2.0).
- Stored as order meta `_wc_other/ns/name` (order/contact) or `_wc_billing/ns/name` and `_wc_shipping/...` (address). Read with `$order->get_meta()`.

Store API `/wp-json/wc/store/v1`

```
woocommerce_store_api_register_endpoint_data( array(
 'endpoint' => 'cart-item', // cart|checkout|product
 'namespace' => 'pocket-coffee',
 'data_callback' => fn( $item ) => array(...),
 'schema_callback' => fn() => array(...),
 'schema_type' => ARRAY_A ) );
woocommerce_store_api_register_update_callback( array(
 'namespace' => 'pocket-coffee', 'callback' => ... ) );
```

- Data appears under `extensions.<namespace>`; write path `POST cart/extensions {namespace,data}` (JS `extensionCartUpdate()`). A throwing callback silently empties the namespace.
- Sessions: `Cart-Token` header (from `GET /cart`) or `Nonce` header; writes without either give 401 `..._missing_nonce`. Flow: `cart/add-item`, `cart/update-customer`, `cart/select-shipping-rate`, `checkout`; `/batch` gives 207.
- Rate limit is off by default (filter `woocommerce_store_api_rate_limit_options` : 25 requests / 10 s, POST only).

Block payment method

```
// PHP: X extends AbstractPaymentMethodType
protected $name = 'my_pay'; // = gateway id
initialize() is_active() get_payment_method_data()
get_payment_method_script_handles()
add_action( 'woocommerce_blocks_loaded', fn() =>
 add_action( 'woocommerce_blocks_payment_method_type_registration',
  fn( $r ) => $r->register( new X() ) ) );
// JS
registerPaymentMethod( { name: 'my_pay', label,
 content, edit, canMakePayment: () => true,
 ariaLabel, supports: { features: [ 'products' ] } } );
```

Plus the `cart_checkout_blocks` declaration. Gateway class: `WC_Payment_Gateway` with `process_payment()`, `process_refund()`, `$this->supports`.

Themes

- **Block theme** (has `templates/index.html`): `theme.json` (schema v3), `templates/`, `parts/`, `patterns/`, `styles/`. Woo slugs: `single-product`, `archive-product`, `taxonomy-product_cat`, `_tag`, `_attribute`, cart and checkout templates. Style blocks such as `woocommerce/product-price` in `theme.json`.
- **Classic theme**: hooks first, overrides last. Copy `plugins/woocommerce/templates/<path>` to `theme/woocommerce/<path>`; the `@version` header goes stale (e.g. `single-product/meta.php` became 11.2.0); WooCommerce > Status lists outdated overrides.
- Defaults: Purple (block), Storefront (classic).

REST v3 and webhooks

```
curl -u "$CK:$CS" "$STORE/wp-json/wc/v3/products?per_page=2&_fields=id,sku,price"
```

- Keys: WooCommerce > Settings > Advanced > REST API (read, write, read/write; the key never exceeds its owner's rights). **HTTPS**: Basic (key = user, secret = password). **HTTP**: OAuth 1.0a one-legged; Basic over HTTP is anonymous (401).
- Apps ask for keys at `/wc-auth/v1/authorize` (`app_name`, `scope`, `user_id`, `return_url`, HTTPS `callback_url`). Errors: 401 `..._authentication_error`, 403 `..._cannot_view`.
- **Webhooks** topics `order.created|updated|deleted|restored`, `product.*` (`product.published` since 10.8.0). Delivery via Action Scheduler, group `woocommerce-webhooks`; POST, 60 s timeout. Verify `X-WC-Webhook-Signature` = base64 HMAC-SHA256 of the raw body; compare with `hash_equals`; de-duplicate (the delivery id is not unique).

`wp wc` commands

```
wp wc product create --name=Mug --type=simple \
 --regular_price=9.50 --sku=PC-MUG --user=admin --porcelain
wp wc product list --sku=PC-MUG --format=csv --user=admin
wp wc product_variation create $PID --regular_price=9.90 ...
wp wc shop_order create --status=processing \
 --customer_id=$CID --payment_method=bacs --user=admin
wp wc shop_order_refund create $OID --amount=5.00 \
 --api_refund=false --user=admin
```

- **Always pass `--user`** (else 401). Commands come from the REST routes: `wp help wc product list`. Formats: table, csv, json, yaml, ids, count.

Caching exclusions

- Never cache **Cart, Checkout, My Account**; bypass non-GET/HEAD, queries `add-to-cart`, `wc-ajax`, `wc-api`, paths `/wp-json/` (including `wc/store/v1` and `wc/v3`), `wp-admin`, `wp-login`, `wp-cron`, `xmlrpc.php`.
- Bypass on cookies `woocommerce_items_in_cart`, `woocommerce_cart_hash`, `wp_woocommerce_session_*` (2 days), logged-in cookies. Do not cache DB rows `_wc_session_`; a response with `Set-Cookie` is not stored.
- Also: persistent object cache, OPcache, system cron with `DISABLE_WP_CRON`.

Security checklist

- Patch WordPress, WooCommerce, plugins and themes fast; remove unused ones. July 2021: SQL injection (CVE-2021-32789, Blocks) fixed by a forced update from WordPress.org; rotate keys after incidents.
- Block XML-RPC (`system.multicall`); restrict user enumeration (`/wp-json/wp/v2/users`, `/?author=1`); delete `readme.html`; `expose_php=Off`.
- Permissions 755 / 644, `wp-config.php` 400 or 440; the web server writes only `uploads`; keep `wc-logs` private.
- `define( 'DISALLOW_FILE_EDIT', true ); define( 'FORCE_SSL_ADMIN', true );`
- One API key per integration, least role, `read` unless writing; revoke unused. Shop manager role for staff; 2FA.

Testing, AI and Marketplace

- PHPCS (`WooCommerce-Core`/`WordPress-Extra`, `PHPCompatibilityWP`), PHPUnit 9.6 with `WP_UnitTestCase`, Playwright (`@woocommerce/e2e-utils-playwright`), QIT, CI. Marketplace: HPOS-compatible, two latest WP/WC versions.
- AI: MCP Adapter added 10.3.0; **11.3 removes the `/wp-json/woocommerce/mcp` endpoint** and stops bundling the adapter.

CLOUD DEPLOYMENT: MANY PODS ON AMAZON EKS (CHECKED 2026-10-29)

Database. RDS for MariaDB 11.8 or MySQL 8.4 sized from orders, not page views; HPOS on and compatibility mode off after `wp wc hpos verify_data`; clean `woocommerce_sessions`; checkout always reads the writer, never a replica.

State placement. Media on S3 behind CloudFront; object cache on Valkey; sessions in the `woocommerce_sessions` table; CloudFront bypasses cart, checkout, account, Store API, `wc-ajax` and `wc-api`; payment webhooks reach any pod and must be idempotent; extensions only through the image.

Cron. `DISABLE_WP_CRON` true plus one CronJob for WP-Cron; Action Scheduler async runner off (`action_scheduler_allow_async_request_runner` false) and a worker Deployment looping `wp action-scheduler run`, scaled by KEDA on pending actions.

Release steps. Pre-upgrade Job runs `wp wc update` once with a migration database user (automatic update off through `woocommerce_enable_auto_update_db`), rolling update, RDS snapshot before; freeze calendar for peak season.

Key commands. `wp wc hpos status`, `wp action-scheduler status`, `wp wc update`, `wp wc webhook list`, `wp maintenance-mode activate`, `helm upgrade pocketcoffee ./pocketcoffee-woocommerce`.