

PrestaShop Cheat Sheet

VERSIONS AND SUPPORT

Branch	Latest	Status 2026-10-27
9.2.x	9.2.0 2026-09-30	maintained until 9.3.0
9.1.x /	9.1.5 08-18 /	ended
9.0.x	9.0.3 02-03	
8.2.x	8.2.8 2026-08-18	extended: critical and security fixes until 10.0.0

PrestaShop	PHP (recommended)	Symfony
9.2 / 9.1	8.1 to 8.5 (8.5)	6.4 LTS
9.0	8.1 to 8.4 (8.4)	
8.x	7.2 to 8.1 (8.1)	4.4

PHP security ends: 8.1 2025-12-31, 8.2 2026-12-31, 8.3 2027-12-31, 8.4 2028-12-31, 8.5 2029-12-31: use 8.4 or 8.5 on new shops. The 9.2 PHP range is inferred.

9.2.0 ships: Symfony 6.4.46, Smarty 4.5.7, Twig 3.27.1, Doctrine ORM 2.20, API Platform 3.4.17, Hummingbird 1.2 (default theme), Classic 3.1.2 (deprecated). Needs MySQL 5.7+ or MariaDB 10.2+, Apache 2.4, memory_limit 512M; Node 24+ builds Hummingbird.

8 to 9: FrameworkBundleAdminController deprecated 9.0, removed 10.0 (use PrestaShopAdminController); legacy product page (18 hooks) and 7 login hooks removed; Guzzle, SwiftMailer gone.

INSTALL AND DOCKER

<pre>docker compose up -d # prestashop/prestashop:9.2.0 = latest # PHP 8.5.12; tags 9.2.0-8.1 .. -8.5, -fpm, nightly php install/index_cli.php --db_server=.. -- db_name=.. --domain=..</pre>	<pre>run index_cli.php on first start; needs DB_SERVER, DB_PORT, DB_NAME, DB_USER, DB_PASSWD host:port seen by the browser (else container IP is stored) first employee ADMIN_MAIL, ADMIN_PASSWD PS_ENABLE_SSL HTTPS PS_DEV_MODE default 0; any string but 0/empty (even "false") = dev ON PS_FOLDER_ADMIN leave default: the installer fails</pre>
9.2 deletes install/ after install; the installer renames admin randomly.	

MODULE SKELETON

<pre>modules/pcfcbasics/ pcfcbasics.php config.xml composer.json src/Install/Installer.php upgrade/upgrade- 1.1.0.php config/{services,routes}.yaml views/{templates/hook,css}</pre>	<pre>if (!defined('_PS_VERSION_')) { exit; } class Pcfcbasics extends Module { // = folder name construct(): name, tab, version, author, need_instance=0, ps_versions_compliancy= ['min'=>'9.0.0', 'max'=>'9.99.99'], bootstrap=true; parent::__construct(); displayName, description, confirmUninstall via \$this->trans() isUsingNewTranslationSystem(): true install(): parent::install() && tables && Configuration && \$this->registerHook('displayHeader') uninstall(): drop config + tables && parent::uninstall() hookDisplayHeader(array \$params) {} // 'hook' + Name</pre>
<p>Composer: "type": "prestashop-module", PSR-4 src/, require vendor/autoload.php. Tables: CREATE TABLE IF NOT EXISTS in install, drop in uninstall.</p> <p>Upgrade script: file upgrade/upgrade-1.1.0.php (exactly one dash: upgrade-1.2.0-rc1.php never runs), function upgrade_module_1_1_0(\$module) , run in order when the file version is newer; install() is not re-run, so add schema, keys and hooks there; anything but true stops the chain.</p>	

CQRS: COMMAND AND QUERY

<pre>CommandBusInterface::handle(\$message); ids prestashop.core.command_bus, prestashop.core.query_bus; both wrap Symfony Messenger (synchronous, one handler per message, domain exceptions arrive unwrapped).</pre>	<pre>#[AsCommandHandler] // or #[AsQueryHandler] final class SetRoastDateHandler { public function invoke() — SetRoastDateCommand \$c): void {} } \$commandBus->handle(new SetRoastDateCommand(\$id, \$date)); \$info = \$queryBus->handle(new GetRoastDate(\$id));</pre>
<pre>prestashop:list:commands-and-queries [-- domain=Tax]; debug:messenger.</pre>	

COMMON HOOKS

Display hooks return HTML; action hooks ignore the return.	
Front	displayHeader, displayHome, displayFooter, displayFooterBefore, displayProductAdditionalInfo, displayProductButtons, displayOrderDetail, displaySearch
Assets	actionFrontControllerSetMedia, actionFrontControllerSetVariables, actionAdminControllerSetMedia
Orders	actionValidateOrder, actionOrderHasBeenShipped, actionOrderHasBeenDelivered, actionCheckoutBuildProcess
Modules	actionModuleEnable, actionModuleDisable, actionModuleUpgradeAfter (9.1)
Dispatch	actionDispatcherBefore, actionDispatcher, actionDispatcherAfter
<pre>Hook::exec(\$hook_name, \$hook_args = [], \$id_module = null, \$array_return = false, \$check_exceptions = true, \$use_push = false, \$id_shop = null, \$chain = false) // own hook: \$h = new Hook(); name, title, description, // position = true; \$h->add(); delete it in uninstall() // chain=true: each subscriber gets the previous result</pre>	
Smarty: {hook h='displayX' product=\$product}.	

SERVICES, ROUTES, ADMIN CONTROLLER

<pre>config/ Symfony: back office + Admin API (not front) config/admin/ Symfony + admin legacy container config/front/ light front container, explicit arguments</pre>	<pre>File priority, first wins (9.2+): services.php, services- 9.2.yaml, services-9.yaml, services.yaml.</pre>
<pre>services: _defaults: { public: false, autorewire: true, autoconfigure: true } PrestaShop\Module\Pcfddadmin: { resource: '../src/*', exclude: ['../src/index.php', '../src/*/index.php'] } # config/routes.yaml pcfddadmin_roasts: _path: /pcfddadmin/roasts _methods: [GET] _defaults: _controller: 'PrestaShop\Module\Pcfddadmin\Controller\RoastContr oller:indexAction' _legacy_controller: AdminPcfddadminRoasts # Tab class_name _legacy_link: AdminPcfddadminRoasts class RoastController extends PrestaShopAdminController { #[AdminSecurity("is_granted('read', request.get('_legacy_controller')")] public function indexAction(Request \$r): Response { return \$this- >render('@Modules/pcfddadmin/views/templates/admin/ x.html.twig', []); } #[DemoRestricted(redirectRoute: 'pcfddadmin_roasts')] // writes // \$this->isCsrfTokenValid('id', \$t); \$this- >addFlash(..)</pre>	
<p>Declare the page in main class \$tabs (class_name, parent_class_name): it creates the menu entry and 4 roles ROLE_MOD_TAB_*_CREATE READ UPDATE DELETE . After services or routes change: cache:clear (and --app-id=admin-api).</p>	

OBJECTMODEL AND Db SAFETY RULES

<ul style="list-style-type: none">Table = <code>_DB_PREFIX_ + module + name</code>; <code>Db::execute() / executeS()</code> add no prefix, <code>insert/update/delete</code> and <code>DbQuery::from()</code> do. Engine <code>_MYSQL_ENGINE_</code>, utf8mb4.Cast and escape: (int) / (float) , pSQL(\$s) strings, bqSQL(\$c) identifiers; pSQL(\$s, true) keeps HTML (purifyfirst).Never alter core tables or add foreign keys to them: own table, ObjectModel or Extra Properties (9.2). Never show SQL errors.	<pre>public static \$definition = ['table'=>'pcf47_roast_batch', 'primary'=>'id_pcf47_roast_batch', 'multilang'=>tr ue, 'fields'=>['sku'=> ['type'=>self::TYPE_STRING, 'validate'=>'isReferenc e', 'required'=>true, 'size'=>32], 'kg'=> ['type'=>self::TYPE_FLOAT, 'validate'=>'isFloat'], 'note'=> ['type'=>self::TYPE_STRING, 'lang'=>true, 'validate' =>'isCleanHtml']];</pre>
<p>Keys: type, validate, required, size, values, range, default, lang, shop, allow, null. Types: INT, BOOL, FLOAT, STRING, HTML, DATE, SQL, NOTHING (unsafe). Overrides are exclusive and forbidden in partner modules: prefer hooks or #[AsDecorator].</p>	

THEME FILES AND theme.yml

<pre>themes/NAME/ config/theme.yml preview.png assets/css/ templates/{catalog,layouts}/*.tpl modules/MOD/views/..</pre>	<pre>parent: hummingbird name: pcfgcoffee (= folder) display_name: Pocket Coffee version: 1.0.0 author: {name,url} meta: {compatibility: {from: 9.2.0, to: ~9.2.0, framework: bootstrap-v5.3.3}, available_layouts: {..}} assets: {use_parent_assets: true, css: {all: [{id: x, path: assets/css/x.css, priority: 210}]}} global_settings: {image_types: {cart_default: {width: 125, height: 125, scope: [products]}, ..}, hooks: {custom_hooks: []}} theme_settings: {default_layout: layout-full- width}</pre>
<p>All parent image_types are required or prestashop:theme:enable fails. Override: {extends file='parent:catalog/product.tpl'} + {block name='x' append}.</p>	

WEBSERVICE (/api): key as Basic user

<pre>Configuration:updateValue('PS_WEBSERVICE', 1); // Apache CGIPassAuth On \$sk = new WebserviceKey(); \$sk->key = bin2hex(random_bytes(16)); \$sk->save(); WebserviceKey::setPermissionForAccount(\$sk->id, ['products'=>['GET'=>1]]); \$sk->associateTo([1]); curl -s -g -u "\${\$PS_WS_KEY}:" "\$S\$/customers? display=id,email &filter[email]=%{example.test}&sort= {id_DESC}&limit=0,10&output_format=JSON" # filters [a] ![a] >[n] [a,b] %[x]; ? schema=blank synopsis # POST (XML) 201, PUT, PATCH, DELETE; Local- Content-Sha1 -> 304</pre>
--

ADMIN API (/admin-api)

<pre>bin/console prestashop:api-client create erp -- name="ERP" \ --scopes=product_read,customer_group_write -- timeout=900 curl -X POST https://shop.example.test/admin- api/access_token \ -d grant_type=client_credentials -d client_id="\$\${PS_CLIENT_ID}" \ -d client_secret="\$\${PS_CLIENT_SECRET}" -d scope="product_read" # { 'token_type': "Bearer", "expires_in": 900, "access_to ken": "...", curl -H "Authorization: Bearer \$\${PS_TOKEN}" \ "https://shop.example.test/admin-api/products? limit=2&orderBy=productId&sortOrder=desc" # POST 201 PUT PATCH partial DELETE 204</pre>	<p>OAuth2 client credentials + JWT. Scopes RESOURCE_read write (71 in 9.2.0); write does not imply read (403). HTTPS only; switch PS_ENABLE_ADMIN_API .</p>
--	---

bin/console AND CACHE

<pre>prestashop:config get set remove KEY [--value=], prestashop:module install uninstall enable disable reset upgrade del ete, prestashop:module:list , prestashop:feature- flag list enable disable, prestashop:api-client create delete, prestashop:employee:create-admin, prestashop:theme:enable export, prestashop:search:index [--full], prestashop:thumbnails:regenerate, prestashop:htaccess:generate, debug:router container messenger.</pre>	<pre>bin/console cache:clear --env=prod [--app- id=admin-api] bin/console cache:warmup --env=prod # -e default is dev! # --no-debug --app-id=admin admin-api -n -vvv; debug:router # failing module action still exits 0: check module:list</pre>
---	---

SECURITY HARDENING CHECKLIST

- Patch core and modules separately (ps_facetedsearch < 4.0.4: CVE-2026-54159, CVSS 10).
- Production: `_PS_MODE_DEV_` , `_PS_DEBUG_PROFILING_` false; PHP 8.4/8.5; HTTPS via `PS_SSL_ENABLED_EVERYWHERE` .
- Core sends no CSP, X-Frame-Options or HSTS: add them at the web server (CSP Report-Only first).
- DB account limited to the shop schema; web user writes only var/, img/, upload/, download/.
- Back office: strong passwords, minimum profiles, admin folder limited by IP/VPN; `PS_SHOP_ENABLE + PS_MAINTENANCE_IP` for maintenance.
- Module code: Smarty {escape: 'html': 'UTF-8'} (avoid nofilter); CSRF tokens + `hash_equals()` ; # [AdminSecurity] on admin actions.

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

<p>Database. RDS for MySQL 8.4 (Aurora MySQL 3); parameters, php reads the environment; TLS through a dbPDO override; separate migration user; checked on PrestaShop 9.2.0 and 8.2.8.</p>	<p>State placement. Shared EFS for img/, upload/, download/ and the theme settings; var/cache and var/logs per pod; sessions in Valkey (session.save_path); SQL cache on Memcached (CacheMemcached); PS_SHARTY_LOCAL 1; the same cookie keys on every pod; admin folder renamed at build; code and modules baked into the image.</p>	<p>Cron. One CronJob (pocketcoffee-cron, concurrencyPolicy: Forbid) calling module cron URLs or bin/console commands; web pods schedule nothing.</p> <p>Release steps. Pre-upgrade Job pocketcoffee-migrate: prestashop:module upgrade <name>; a core upgrade is an image rebuild plus a database upgrade; cache:warmup --env=prod per pod; maintenance mode; RDS snapshot first.</p>	<p>Key commands. php bin/console prestashop:config get PS_SSL_ENABLED --env=prod, prestashop:module:list, cache:warmup --env=prod, kubectl -n pocketcoffee rollout status deployment/pocketcoffee-web, helm upgrade pocketcoffee ./pocketcoffee-prestashop.</p>
--	---	---	--