Developer path

A path for the developer who builds or extends an Exponential store - the code map of the shop, the extension points with their settings, the RAD wizards that generate handler skeletons, and the order in which to read...

Source file: `doc/eng-us/help/e-commerce/roles/developer-path.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/e-commerce/roles/developer-path.md

This page is for developers who build an Exponential store or extend the shop: in-house developers and the developers of an agency. It gives the code map of the shop, every extension point with the setting that selects it, and a reading order through the technical pages.

Contents

1. The code map
2. Extension points at a glance
3. Generators that save time
4. Reading order
5. Rules of thumb

The code map

What	Where
Shop module: views, policy functions, limitations	`kernel/shop/module.php`
The views' code (entry files call classes)	`kernel/shop/*.php`, `kernel/private/classes/views/shop/`
Shop operations (what triggers exist and what runs in between)	`kernel/shop/operation_definition.php`, `kernel/shop/ezshopoperationcollection.php`
Template fetch functions (`fetch( 'shop', ... )`)	`kernel/shop/function_definition.php`, `kernel/shop/ezshopfunctioncollection.php`
Orders, items, statuses, status history	`kernel/classes/ezorder.php`, `ezorderitem.php`, `ezorderstatus.php`, `ezorderstatushistory.php`
Baskets and product collections	`kernel/classes/ezbasket.php`, `ezproductcollection.php`, `ezproductcollectionitem.php`, `ezproductcollectionitemoption.php`
VAT	`kernel/classes/ezvattype.php`, `ezvatrule.php`, `ezvatmanager.php`, `vathandlers/`
Shipping and basket info	`kernel/classes/ezshippingmanager.php`, `basketinfohandlers/`
Shop accounts and the order mail	`kernel/classes/ezshopaccounthandler.php`, `shopaccounthandlers/`, `ezconfirmorderhandler.php`, `confirmorderhandlers/`
Discounts and wishlist	`kernel/classes/ezdiscount*.php`, `ezuserdiscountrule.php`, `ezwishlist.php`

What	Where
Payment base classes	`kernel/shop/classes/ezpaymentgateway.php`, `ezredirectgateway.php`, `ezpaymentobject.php`, `ezpaymentcallbackchecker.php`
Workflow event types used by the shop	`kernel/classes/workflowtypes/event/ezpaymentgateway/`, `ezsimpleshipping/`, `ezmultiplexer/`
Currencies and exchange rates	`kernel/shop/classes/ezcurrencydata.php`, `ezcurrencyconverter.php`, `exchangeratehandlers/`
Shop templates of the standard design	`design/standard/templates/shop/`

Extension points at a glance

You want to	Extension point	Selected by	Page
Charge shipping by your own rules	Shipping handler	`shop.ini [ShippingSettings] Handler`	Custom shipping handler
Decide VAT by your own rules	VAT handler	`shop.ini [VATSettings] Handler`	Custom VAT handler
Take payment with another provider	Payment gateway	`paymentgateways.ini`, a payment gateway event	Payment gateway development
Collect different customer data	Shop account handler	`shopaccount.ini [AccountSettings] Handler`	Shop account handlers
Change the order mail	Confirm order handler	`shopaccount.ini [ConfirmOrderSettings] Handler`	Basket info and confirm order handlers
Change basket totals	Basket info handler	`shop.ini [BasketInfoSettings] Handler`	same
Fetch exchange rates elsewhere	Exchange rates update handler	`shop.ini [ExchangeRatesSettings] ExchangeRatesUpdateHandler`	Exchange rate handlers
A price field of your own	Price datatype	`site.ini [ShopSettings] ProductDatatypeStringList[]`	Custom price datatypes
Run code at a shop step	Workflow event type on a shop trigger	`workflow.ini`	Workflows on shop triggers

Generators that save time

The RAD tools in **Setup** write working skeletons into an extension:

- `setup/handlerextension/<kind>` (policy `setup/setup`) has recipes for a payment gateway (redirect and transparent), a VAT handler, a shipping handler, a basket info handler and an exchange rate handler, each with the methods to write and the setting it needs (Extensions).
- `setup/workflowevent` writes a workflow event type (Writing a workflow event type).

Check a generated file against the loader paths described on each extension page before you rely on it, and run `php bin/php/checkclasses.php --quiet-ok` afterwards.

Reading order

1. Running a shop: overview and Basket, checkout and orders: the parts.
2. Checkout operations step by step and Temporary orders and activation: what runs when.
3. Shop settings walkthroughs: every setting.
4. Workflows on shop triggers and Payment gateway event.
5. The page of the extension point you need in Extending the shop.
6. Testing a store and Troubleshooting checkout.

Rules of thumb

- Put shop settings in `settings/override/` or a siteaccess folder, never in the shipped `settings/*.ini` (Shop settings walkthroughs).
- Put handlers and gateways in an extension of your own, not in `kernel/` (Packaging as an extension).
- Never log or store card data; the audit trail records payments without it, and so should your code (Security for applications).
- Test every change with a full order in the provider's test mode (Testing a store).

Related pages

- Paths by role
- App creators: start here
- Workflows, triggers and events
- Headless commerce with expservices
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.