Glossary for PHP developers

The terms a PHP developer meets in Exponential code and documentation, from accumulator and append.php to worker and zygote, each with a one-line meaning and the page or file that explains it.

Source file: `doc/eng-us/help/php-developers/glossary-for-php-developers.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/php-developers/glossary-for-php-developers.md

This page is for PHP developers who meet a term in Exponential code or documentation and want its meaning in one line. Terms are in alphabetical order; each names where to read more.

Term	Meaning
accumulator	an eZDebug timer that adds up the time and calls of a piece of code across a request (Debugging with eZDebug)
append.php file	an INI file named `*.ini.append.php`, wrapped in a PHP comment so the web server never shows it (Reading settings with eZINI)
`autoload.php`	the first file of every entry point: defines EXP_ROOT_DIR, loads PHP compatibility functions, registers the class map autoloader (How a request runs in PHP)
baseline (PHPStan)	`phpstan-baseline.neon`, the findings accepted when the check started; only new findings fail (Static analysis with PHPStan)
class map	a generated array of class name to file: `autoload/ezp_kernel.php`, `var/autoload/ezp_extension.php` (Autoloads and class maps)
`cleanExit()`	`eZExecution::cleanExit()`, ends a request that sent its own output; under Velocity by an exception
content class, object, node	the type of a piece of content, the content with its versions and languages, one location of it in the tree (The content API in PHP)
cronjob part	a named group of scripts in `cronjob.ini` run by `runcronjobs.php <part>` (Write a cronjob part)
datatype	the PHP class behind a field type of content classes, extending `eZDataType` (Write a datatype)
`.dba` file	a database schema or data file that the schema handler turns into DDL for each engine
debug condition	a switch in `debug.ini [GeneralCondition]` for `eZDebugSetting::writeDebug()` messages
design chain	the ordered list of designs searched for a template, ending at `standard`
deploy (`exp:velocity deploy`)	one command that regenerates autoloads, clears caches, reloads PHP-FPM and restarts Velocity in order
engine (Velocity)	the program that serves the site: `qbix` (recommended), `frankenphp` or `php` (Velocity engines)
engine archive	`dist/engine.phar`, an optional archive of `kernel/`, `lib/` and `autoload/` (The engine archive (phar) for PHP developers)
EXP_ENV	the environment name of a machine, in `config.env.php`; selects INI variants such as `site.ini.dev.append.php`
EXP_ROOT_DIR	the installation root as an absolute path, defined by `autoload.php`

Term	Meaning
exp prefix	the prefix of new project code; eZ identifiers of the kernel are never renamed (Class naming and the exp prefix)
expservices	an extension of service classes over ezjscore with discovery, an envelope and API tokens
extension	a directory under `extension/` with its own settings, classes, modules, designs; the unit of everything added to the kernel
extension point	a place where code plugs in without editing the kernel, listed in **Setup > RAD** (Extension points from PHP)
`ezjscore`	the extension that calls registered static PHP methods over HTTP and answers in JSON (ezjscore JSON endpoints in PHP)
ezpEvent	the kernel's event hub: `notify()` and `filter()` events, listeners in `site.ini [Event]` (Events, listeners and filters)
fetch function	a PHP method templates call with `fetch( 'module', 'function', hash( ... ) )`, declared in `function_definition.php` (Write a fetch function)
filter event	an event whose listeners receive a value and must return it
form token	the CSRF token `ezformtoken` adds to POST forms and checks on POST (CSRF protection and ezformtoken)
handler	a class named by a setting and loaded with `eZExtension::getHandlerClass()`
kernel override	replacing a kernel class through `var/autoload/ezp_override.php`; the last resort
layer (settings)	one of the places an INI value can come from: defaults, extensions, siteaccess, overrides
manifest	`share/filelist.md5`, the checksums the upgrade check compares files with (Git hooks and make targets)
module, view	the first and second part of a system address (`/module/view/...`), declared in `module.php` (Write your first module view)
pagelayout	the template around every page's content (`pagelayout.tpl`)
persistent object	a class extending `eZPersistentObject` that maps a table (Persistent object definitions in depth)
`phpcompat.php`	`lib/phpcompat.php`, PHP functions newer than 8.0 defined only where PHP lacks them
policy, limitation	a permission (`module/function`) of a role, and the restriction on it (section, subtree, class)
request key	`eZExecution::requestKey()`, the request's `REQUEST_TIME_FLOAT`, for once-per-request work
runnable	a class form of a command, cronjob part or module view, extending `Exponential\Runnable\Runnable` (Module views as classes)
service (kernel)	a class that a view, a command and a cronjob part share, such as `Exponential\Service\Trash`
siteaccess	a named set of settings selected by host, port or path; decides design, language and database
static state	values in static properties, function statics or globals that survive the request in some situations (Static state traps)

Term	Meaning
template operator	a PHP class the template engine calls for `{$value|operator( ... )}` (Write a template operator in PHP)
trigger	the connection of a workflow to an operation, such as `content/publish/before`
Velocity	Exponential's application server with persistent workers, part of Exponential as of 6.0.15 (Velocity persistent workers for PHP developers)
warm-up	Velocity's first boot of the kernel, inherited by the workers
wash	the template operator that escapes a value for HTML
worker	a Velocity process that serves many requests
workflow event type	a step administrators can put into workflows, a class extending `eZWorkflowEventType` (Write a workflow event type)
zygote	the Velocity process from which new workers are forked, holding no visitor connection

Related pages

- Glossary for extension developers
- PHP developer FAQ
- PHP Developers: start here
- 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.