Composer for PHP developers

How Exponential uses Composer - the root composer.json with PHP 8.0 to 8.5 (PHP 7.4 usable without support), required extensions and packages, the legacy-scripts hook that regenerates autoloads, the installer plugin...

Source file: `doc/eng-us/help/php-developers/composer-for-php-developers.md` Last changed 2026-11-07
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/composer-for-php-developers.md

This page is for PHP developers who add a dependency, package an extension or read the root `composer.json`. Exponential installs with Composer from Packagist, but it is not a PSR-4 Composer project: Composer installs the extensions and the libraries, and the kernel's own class maps load the classes.

Contents

1. The root composer.json
2. How extensions are installed
3. The development tools
4. An extension's composer.json
5. Version constraints
6. Rules

The root composer.json

Part	Holds
`name`, `type`	`se7enxweb/exponential`, type `ezpublish-legacy`
`license`	`GPL-2.0-or-later`
`require php`	`^8.0 || ^8.1 || ... || ^8.5` (and forward-looking ranges to 8.8)
`require ext-*`	`dom`, `libxml`, `mbstring`, `pcre`, `json`, `iconv`, `reflection`, `session`, `spl`, `simplexml`
`require` packages	the installer plugin, Velocity (`se7enxweb/exponential-velocity`), the shipped `se7enxweb/*` extensions, the kernel's libraries `se7enxweb/expcomponents-*` (the expcomponents, Exponential's fork of the Zeta Components, since 6.0.15)
`require-dev`	`phpunit/phpunit` (`^9.6 || ^10.5 || ^11.5 || ^12.0 || ^13.4`)
`suggest`	database extensions (`ext-sqlite3`, `ext-pgsql`, `ext-mysqli`, `mongodb/mongodb`), `ext-gd`, `ext-curl`, optional extensions
`autoload`	`files:` `autoload.php`, so Composer's autoloader loads the kernel's
`scripts`	`legacy-scripts` runs `@php bin/php/ezpgenerateautoloads.php` after `install` and `update`

How extensions are installed

Packages of type `ezpublish-legacy-extension` are put into `extension/<name>/` (not `vendor/`) by the installer plugin `se7enxweb/exponential-legacy-installer`, which `config.allow-plugins` permits. After the install the `legacy-scripts` hook regenerates the class maps, so the extension's classes are found. Libraries (`se7enxweb/expcomponents-*`, `mongodb/mongodb`) go to `vendor/` and are loaded by Composer's own autoloader.

The development tools

PHP_CodeSniffer, PHPCompatibility and PHPStan are **not** in the root `require-dev`. They live in `composer.dev.json`, resolved for PHP 8.0.30 and installed into `.devtools/vendor` by `make devtools` (`COMPOSER=composer.dev.json composer install`), so they never touch the site's `vendor/` (Git hooks and make targets).

An extension's composer.json

JSON

```
{
    "name": "acme/acmereport",
    "description": "Monthly reports for Exponential",
    "type": "ezpublish-legacy-extension",
    "license": "GPL-2.0-or-later",
    "require": {
        "php": "^8.0",
        "se7enxweb/exponential-legacy-installer": "*"
    }
}
```

- `type` makes the installer put it into `extension/`; the directory name is the part of the package name after the `/`, unless `extra.ezpublish-legacy-extension-name` names another one (`LegacyExtensionInstaller::getInstallPath()`).
- Declare `php` `^8.0` unless you need more (Supported PHP versions: 8.0 to 8.5).
- Declare the libraries your code uses; do not ship `vendor/` inside the extension.
- Keep `ezinfo.php` and `extension.xml` versions equal to the release (composer.json for extensions, Composer packages and Packagist).

Version constraints

Exponential packages use three- and four-part versions (`6.0.15`, `0.0.4.54`). The tilde floats the last position:

Constraint	Allows
`~6.0.15`	`>=6.0.15 <6.1.0`
`~0.0.4.57`	`>=0.0.4.57 <0.0.5.0`
`^8.0`	`>=8.0 <9.0`

After `.9` comes `.10` in the last position (`0.0.4.9`, `0.0.4.10`). Packagist compares each position as a number; a plain lexical sort of tags does not, so use `sort -V` or `git tag`

`--sort=version:refname` when a script picks the newest tag (Versioning rules).

Rules

- **A published version never changes.** A tag on Packagist is read once and cached; fix a mistake with the next version (Tags are permanent).
- **Check Packagist before repinning.** CI's PHPUnit job resolves dependencies fresh; a constraint pointing to a release that is not on Packagist yet fails every leg.
- **Do not edit** `vendor/`, and be careful with `composer update` on an installed site: some installations keep extensions as live git checkouts that Composer would replace.

Related pages

- PHP, PHP-FIG and Composer standards
- Modernizing composer.json
- The Packagist webhook
- 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.