Testing: PHPUnit, database tests and checks

How to test an application on Exponential for developers used to WebTestCase, Laravel's HTTP tests, Django's test client or Rails system tests - PHPUnit without a database for services, tests against a throwaway…

Source file: `doc/eng-us/help/web-framework-developers/testing.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/web-framework-developers/testing.md

This page is for developers who test with a framework's test kit: Symfony's `WebTestCase`, Laravel's HTTP tests and factories, Django's test client and fixtures, Rails system tests. Exponential's tests are plain PHPUnit. The kernel runs most of its suites **without a database**, a few against a throwaway one, and adds checks for templates, settings and links. Do the same in your extension (Testing applications).

Contents

1. The layers
2. Unit tests without a database
3. Tests with a database
4. Templates, settings and pages
5. The kernel's CI as a model
6. What has no built-in equivalent

The layers

Layer	Tool	Catches
Syntax	`php -l`, `bin/php/eztemplatecheck.php`	parse errors in PHP and templates
Unit	PHPUnit, no database	logic of services, validators, handlers, operators
Integration	PHPUnit with `tests/runtests.php --dsn ...`	SQL, persistent objects, content operations
Rendered pages	HTTP requests, browser automation	templates, permissions, redirects
Accessibility and CSP	an accessibility checker, the CSP report log	barriers, inline code

Unit tests without a database

`phpunit.xml` loads `tests/bootstrap.php`, which loads Composer's autoloader and the test toolkit but does not boot the kernel (DB-less PHPUnit tests):

PHP

```php
<?php
// extension/expdemo/tests/expDemoColorValidatorTest.php
class expDemoColorValidatorTest extends PHPUnit\Framework\TestCase
{
    public function testAcceptsHexColors()
    {
        $this->assertTrue( expDemoColorValidator::isValid( '#a1b2c3' ) );
    }
}
```

Shell

```shell
vendor/bin/phpunit extension/expdemo/tests/
```

This works when the logic lives in plain classes with their dependencies passed in (Dependency injection and services). Keep database access out of those classes or behind an interface you can replace in the test.

Tests with a database

`tests/runtests.php --dsn <dsn> --db-per-test` runs suites against a throwaway database, as the kernel's CI job does on MySQL and PostgreSQL. Mark such tests `@group database`; `vendor/bin/phpunit` alone excludes that group. Never point tests at a database that holds real content (PHPUnit in the testing standard).

The test toolkit in `tests/toolkit/` has the base classes `ezpTestCase` and `ezpDatabaseTestCase`, and `ezpINIHelper::setINISetting()` / `restoreINISettings()` to change a setting for one test.

There are no model factories: create the content your test needs with `eZContentFunctions::createAndPublishObject()` and remote IDs, and your own rows with `eZPersistentObject::store()`.

Templates, settings and pages

Shell

```shell
php bin/php/eztemplatecheck.php -s<siteaccess> extension/expdemo/design/   # every template of yo
ur extension
./console exp:ini where expdemo.ini/ListSettings/DefaultLimit <siteaccess>  # settings resolve as
 expected
```

For pages, send HTTP requests to a running installation (PHP's built-in server through Velocity's `php` engine, or a test host) as anonymous, as a restricted user and as an editor. Use a dedicated test user created for the run, never a shared administrator account. Check the browser console and `var/log/csp.log` for blocked inline code.

The kernel's CI as a model

The kernel's `phpunit.xml` has 23 suites; GitHub Actions run them on PHP 8.1 to 8.5 without a database and on MySQL and PostgreSQL with one, plus syntax, PHPStan, code style and a PHP 8.0 boot test. `tests/test-map.json` maps source files to their tests, so a change runs only its own tests before a push (Testing and CI standard, Continuous integration for extensions).

What has no built-in equivalent

Framework feature	In Exponential
A kernel test client that handles a request in-process (`WebTestCase`, `$this->get()`, Django `Client`)	none; test services directly and pages over HTTP
Model factories, fixtures loaders	content API calls and `.dba` seed rows
Transaction rollback per test	`--db-per-test` with a throwaway database
Mail fakes	the `file` mail transport in a test environment variant (Environments)

Related pages

- Testing applications
- Quality checks
- PHPUnit versions
- 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.