Database and live tests

Tests that need a real database or server - the database group and tests/runtests.php with a DSN and a database per test, the MongoDB and live HTTP tests and their environment variables, how to pass passwords safely…

Source file: `doc/eng-us/help/php-developers/database-and-live-tests.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/database-and-live-tests.md

This page is for PHP developers who write or run tests that need a database, a MongoDB server or a running site. Those tests are kept out of the normal PHPUnit run, so that `vendor/bin/phpunit` passes on any laptop and CI runner, and are run separately against servers you name.

Contents

1. Which tests are live
2. Database tests with runtests.php
3. Writing a database test
4. MongoDB and MySQL integration tests
5. Live HTTP tests
6. Passing passwords safely
7. In CI

Which tests are live

Kind	Marked by	Excluded from the normal run
database tests (legacy toolkit)	`@group database`, classes extending `ezpDatabaseTestCase`	yes
MongoDB and MySQL integration	`@group mongodb-live`	yes
real mail	`@group mail-live`	yes
network	`@group network-live`	yes
live HTTP against a site	environment variables; skipped when unset	they skip

A live test reads where its server is from environment variables and skips, naming the variable, when it is not set. No test carries a host name, an account or a password of its own.

Database tests with runtests.php

The legacy runner `tests/runtests.php` creates the schema in a database you name, optionally one database per test, and runs the database group:

Shell

```
php -d date.timezone=America/Los_Angeles tests/runtests.php --dsn mysql://user@127.0.0.1/testdb --db-per-test
php tests/runtests.php --dsn postgresql://user@127.0.0.1/testdb --db-per-test
```

Use a throwaway database and an account that may create and drop databases. Never point it at a site's database: the runner rebuilds the schema.

Writing a database test

The toolkit in `tests/toolkit/` provides `ezpDatabaseTestCase` (`tests/toolkit/ezpdatabasetestcase.php`), which sets up the database connection for a test case:

PHP

```php
<?php
/**
 * @group database
 */
class acmeReportStorageTest extends ezpDatabaseTestCase
{
    public function testStoreAndFetch()
    {
        $item = new acmeReportItem( array( 'status' => 'open', 'amount' => 2 ) );
        $item->store();
        $fetched = acmeReportItem::fetch( $item->attribute( 'id' ) );
        $this->assertSame( 'open', $fetched->attribute( 'status' ) );
    }
}
```

Keep database tests few and focused on what really needs the database (SQL, transactions, storage), and test the logic around them without one (PHPUnit without a database).

MongoDB and MySQL integration tests

Tests	Variables
`expMongoDBIntegrationTest`, `sevenxMongoDBIntegrationTest` (MongoDB part)	`EXP_TEST_MONGO_URI` (a `mongodb://` connection string), `EXP_TEST_MONGO_DATABASE` (default `exp`)
the same tests (MySQL part)	`EXP_TEST_MYSQL_USER`, `EXP_TEST_MYSQL_PASSWORD`, `EXP_TEST_MYSQL_DATABASE`, `EXP_TEST_MYSQL_HOST` (default `localhost`), `EXP_TEST_MYSQL_PORT` (default 3306)
`expMongoDBDeclaredDefaultTest`, `TrashListPagedTest`	`EXP_TEST_MONGO_SCRATCH` (`<host>:<port>/<database>`, a database whose name contains "scratch" or "test")

Shell

```
php vendor/bin/phpunit --testsuite mongodb-live
```

Live HTTP tests

Tests such as `ApiKeyLiveTest` and `RestContentPermissionLiveTest` call a running site over HTTP. They read `EXP_TEST_PUBLIC_HOST` (else the host of the default siteaccess's `SiteURL`), `APIKEY_LIVE_BASES` and `APIKEY_LIVE_ADMIN_PASSWORD`. Run them against a test installation with a test account, never against production and never with a shared administrator account.

A live test that checks what the web server recorded in the audit trail reads the site's audit directory, `expAuditConfig::siteLogDir()`, not the run's own directory under `var/tmp` that `tests/bootstrap.php` sets up.

Passing passwords safely

Pass a password through the environment, read from a file only you can read, not on the command line where it lands in the shell history and the process list:

Shell

```
EXP_TEST_MYSQL_PASSWORD="$(cat ~/.exp-test-mysql-password)" EXP_TEST_MYSQL_USER=test EXP_TEST_MYS
QL_DATABASE=testdb \
    php vendor/bin/phpunit tests/tests/lib/ezdb/mongodb/expMongoDBIntegrationTest.php
```

In CI

The PHPUnit workflow runs **Database tests** on MySQL 8.0 and PostgreSQL 15 (PHP 8.3, with `pgcrypto` on PostgreSQL), using `tests/runtests.php --dsn ... --db-per-test`. SQLite is covered by driver unit tests in `tests/tests/lib/ezdb/`, MongoDB by adapter unit tests without a server; the `mongodb-live` group is not run in CI (Database support).

Related pages

- Quality checks: tests that need a live server
- Code for every supported database
- CI and reproducing a red check
- 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.