Testen: PHPUnit, Datenbanktests und Prüfungen

Wie Sie eine Anwendung auf Exponential testen, für Entwickler, die WebTestCase, die HTTP-Tests von Laravel, den Test-Client von Django oder die Systemtests von Rails gewohnt sind - PHPUnit ohne Datenbank für Dienste...

Source file: `doc/ger-de/hilfe/web-framework-entwickler/testen.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/web-framework-developers/testing.md

Diese Seite richtet sich an Entwickler, die mit dem Testwerkzeug eines Frameworks testen: `WebTestCase` von Symfony, HTTP-Tests und Factories von Laravel, Test-Client und Fixtures von Django, Systemtests von Rails. Die Tests von Exponential sind reines PHPUnit. Der Kernel führt die meisten seiner Suites **ohne Datenbank** aus, einige wenige gegen eine Wegwerf-Datenbank, und ergänzt Prüfungen für Templates, Einstellungen und Links. Machen Sie es in Ihrer Erweiterung genauso (Anwendungen testen).

Inhalt

1. Die Ebenen
2. Unit-Tests ohne Datenbank
3. Tests mit Datenbank
4. Templates, Einstellungen und Seiten
5. Die CI des Kernels als Vorbild
6. Was keine eingebaute Entsprechung hat

Die Ebenen

Ebene	Werkzeug	Findet
Syntax	`php -l`, `bin/php/eztemplatecheck.php`	Parse-Fehler in PHP und Templates
Unit	PHPUnit, ohne Datenbank	die Logik von Diensten, Validatoren, Handlern, Operatoren
Integration	PHPUnit mit `tests/runtests.php --dsn ...`	SQL, persistente Objekte, Inhaltsoperationen
Gerenderte Seiten	HTTP-Requests, Browserautomatisierung	Templates, Berechtigungen, Weiterleitungen
Barrierefreiheit und CSP	ein Prüfwerkzeug für Barrierefreiheit, das CSP-Berichtsprotokoll	Barrieren, Inline-Code

Unit-Tests ohne Datenbank

`phpunit.xml` lädt `tests/bootstrap.php`, das den Autoloader von Composer und das Test-Toolkit lädt, den Kernel aber nicht startet (PHPUnit-Tests ohne Datenbank):

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/
```

Das funktioniert, wenn die Logik in einfachen Klassen liegt, denen ihre Abhängigkeiten übergeben werden (Dependency Injection und Dienste). Halten Sie Datenbankzugriffe aus diesen Klassen heraus oder hinter einer Schnittstelle, die Sie im Test ersetzen können.

Tests mit Datenbank

`tests/runtests.php --dsn <dsn> --db-per-test` führt Suites gegen eine Wegwerf-Datenbank aus, so wie es der CI-Job des Kernels auf MySQL und PostgreSQL tut. Kennzeichnen Sie solche Tests mit `@group database`; `vendor/bin/phpunit` allein schließt diese Gruppe aus. Richten Sie Tests niemals auf eine Datenbank, die echte Inhalte enthält (PHPUnit im Teststandard).

Das Test-Toolkit in `tests/toolkit/` enthält die Basisklassen `ezpTestCase` und `ezpDatabaseTestCase` sowie `ezpINIHelper::setINISetting()` / `restoreINISettings()`, um eine Einstellung für einen einzelnen Test zu ändern.

Model-Factories gibt es nicht: Legen Sie die Inhalte, die Ihr Test braucht, mit `eZContentFunctions::createAndPublishObject()` und Remote-IDs an, und Ihre eigenen Zeilen mit `eZPersistentObject::store()`.

Templates, Einstellungen und Seiten

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
```

Für Seiten senden Sie HTTP-Requests an eine laufende Installation (den eingebauten Server von PHP über die Engine `php` von Velocity oder einen Testhost), als anonymer Besucher, als eingeschränkter Benutzer und als Redakteur. Verwenden Sie einen eigens für den Lauf angelegten Testbenutzer, niemals ein gemeinsam genutztes Administratorkonto. Prüfen Sie die Browserkonsole und `var/log/csp.log` auf blockierten Inline-Code.

Die CI des Kernels als Vorbild

Die `phpunit.xml` des Kernels hat 23 Suites; GitHub Actions führt sie auf PHP 8.1 bis 8.5 ohne Datenbank und auf MySQL und PostgreSQL mit Datenbank aus, dazu Syntax, PHPStan, Codestil

und einen Starttest unter PHP 8.0. `tests/test-map.json` ordnet Quelldateien ihren Tests zu, sodass eine Änderung vor einem Push nur ihre eigenen Tests ausführt (Standard für Tests und CI, Continuous Integration für Erweiterungen).

Was keine eingebaute Entsprechung hat

Funktion im Framework	In Exponential
Ein Kernel-Test-Client, der einen Request im Prozess verarbeitet (`WebTestCase`, `$this->get()`, Django `Client`)	keiner; testen Sie Dienste direkt und Seiten über HTTP
Model-Factories, Fixture-Loader	Aufrufe der Inhalts-API und Startdatenzeilen in `.dba`
Rollback der Transaktion pro Test	`--db-per-test` mit einer Wegwerf-Datenbank
Mail-Attrappen	der Mail-Transport `file` in einer Testvariante der Umgebung (Umgebungen)

Verwandte Seiten

- Anwendungen testen
- Qualitätsprüfungen
- PHPUnit-Versionen
- Freie Dokumentation: Diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

Werden Sie Exponential-Partner

Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.

Partner: hier beginnen | Wie Sie sich bewerben

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.