SQLite database support

Source file: doc/eng-us/features/6.0/sqlite-database.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/features/6.0/sqlite-database.md

This page is for anyone who wants to try Exponential, develop on a laptop, run automated tests, or host a small site on a plan that offers no MySQL. With SQLite, a complete Exponential site runs from one file on disk, with no database server to install, configure or secure.

> **Warning**
>
> `.htaccess` **slows down every request.** When `.htaccess` files are enabled (`AllowOverride` anything but `None`), let alone used, Apache looks for them and reads and evaluates them on every request, in every directory of the path, and page loading times go up drastically (measured). The only real fix is to switch the feature off for the virtual host: `AllowOverride None` in the HTTPS virtual host, and the rules in the virtual host's own configuration as mod_rewrite rules. That is far faster in production speed tests and well worth the time it takes to optimize a production configuration. Switch over right away: copy the rules of the example HTTPS virtual host into your virtual host (with the static cache, also `staticcache_offload.conf`), set `AllowOverride None`, run `apachectl configtest`, reload Apache and test the site. The same rules for other servers: nginx, LiteSpeed, Caddy, IIS, Velocity (built in); where virtual host rules go in a control panel: Plesk, cPanel, DirectAdmin, ISPConfig. **On LiteSpeed or OpenLiteSpeed,** the server's own page cache can also answer anonymous visitors and crawlers without PHP: Page caching with LiteSpeed and OpenLiteSpeed.

SQLite support arrived in January 2024 (releases 6.0.1 to 6.0.3, see the January 2024 chronicle) and has been refined since. Everything below describes the code as it is now; a note says where a setting came in a later release than the first driver.

> **Supported databases (Exponential 6.0.x), in order of preference:** 1. SQLite 3 (the default) · 2. PostgreSQL · 3. MongoDB · 4. MariaDB · 5. Oracle · 6. MySQL. Microsoft SQL Server is known to be possible with additional work at this time. Details: Database support.

What you get

- A kernel database driver, `eZSQLite3DB` (`lib/ezdb/classes/ezsqlite3db.php`), selected with the implementation alias `sqlite3`.
- A schema handler, `eZSQLiteSchema` (`lib/ezdbschema/classes/ezsqliteschema.php`), so the setup wizard, the package installer and schema tools such as `bin/php/ezsqldumpschema.php` and `bin/php/ezsqlinsertschema.php` can create and read tables.
- SQLite-flavoured SQL files in `kernel/sql/sqlite/` (`schema.sql`, `cleandata.sql`), used by the setup wizard for a clean install.
- A setup wizard that lists SQLite as a database choice, tests that the PHP `sqlite3` extension is loaded and explains what to do if it is not.

Try it in five minutes

1. Check that PHP has the extension:

   Shell

   ```
   php -m | grep -i sqlite3
   ```

   It must print `sqlite3`. If it prints nothing, install your distribution's `php-sqlite3` package (or enable `extension=sqlite3` in `php.ini`) and reload PHP-FPM.

2. Open the setup wizard in the browser (`/setup`). On the database page choose **SQLite**. The wizard preselects it when the PHP `sqlite3` extension is present (setting `[DatabaseSettings] DefaultType=sqlite3` in `settings/setup.ini`, see below). If the extension is missing the wizard says so and falls back to the first available database system.

3. Give the database a name (the wizard proposes `sqlite.db`, constant `SQLITE_DEFAULT_FILE_NAME`). A name without a path is created as a file in `var/storage/sqlite3/`:

   Shell

   ```
   ls -l var/storage/sqlite3/
   ```

4. Finish the wizard as usual. The site now runs from that single file.

Settings

File	Block	Key	Default	Scope	What it does
`settings/site.ini`	`[DatabaseSettings]`	`DatabaseImplementation`	`ezmysqli`	siteaccess	Set to `sqlite3` to use the driver.
`settings/site.ini`	`[DatabaseSettings]`	`Database`	(site specific)	siteaccess	The database file. A name without a slash lives in `var/storage/sqlite3/`; an absolute path (starting with `/`) is used as it is; `:memory:` is a throw-away in-memory database.
`settings/site.ini`	`[DatabaseSettings]`	`ImplementationAlias[sqlite3]`	`eZSQLite3DB`	global	Maps the alias to the driver class.
`settings/dbschema.ini`	`[SchemaSettings]`	`SchemaHandlerClasses[sqlite3]`	`eZSQLiteSchema`	global	Schema handler used by installers and schema tools.
`settings/setup.ini`	`[DatabaseSettings]`	`DefaultType`	`sqlite3`	global	Database system the wizard lists first. One of `sqlite3`, `mysqli`, `pgsql`, `mongodb`.
`settings/site.ini`	`[DatabaseSettings]`	`SQLitePragmas[]`	empty list	siteaccess	Extra `name=value` PRAGMAs applied on every connection (later releases).

File	Block	Key	Default	Scope	What it does
settings/site.ini	[DatabaseSettings]	SQLiteTransactionWait	25	siteaccess	Seconds a transaction waits at its start for the other writers before giving up with "database is busy" (later releases). Keep it below the request timeout: Velocity ends a request after Q.webserver.requestTimeout, 30 s unless changed, hence 25; under PHP-FPM the limit is the pool's request_terminate_timeout. Raise both together.

The driver's built-in PRAGMA defaults (also later releases) are `synchronous=NORMAL`, `cache_size=-65536` (64 MB), `mmap_size=268435456`, `temp_store=MEMORY` and `busy_timeout=5000`, and the journal runs in WAL mode. See the driver specification.

Using a database file that another application shares

Since April 2026 (`cefe8f13a2`) the driver accepts an absolute path in `Database`, and registers the `eZSQLite3DB` class in the kernel autoload array. That lets a legacy bridge point Exponential at the very file a Symfony/Doctrine application uses, so both stacks see the same data:

INI

```ini
# settings/siteaccess/<name>/site.ini.append.php
[DatabaseSettings]
DatabaseImplementation=sqlite3
Database=/absolute/path/to/shared.db
```

The directory of a new database is created recursively (`ae5feaf6a7`), so a nested path works on first use.

Limits and good practice

- SQLite allows one writer at a time. Many editors publishing at the same moment queue behind each other (see `SQLiteTransactionWait`); for a busy editorial team choose MySQL, PostgreSQL or MongoDB.
- Do not put the database file inside a directory the web server serves directly. In the shipped `.htaccess` only a short list of `var/` paths (public images, public caches, preview images of packages) is served as files; the rest goes to `index.php`. Verify on your own host that the database file is not downloadable: `curl -sI https://www.example.com/var/storage/sqlite3/sqlite.db`. A status other than 200 is what you want.
- Run `ANALYZE` once after a big import so the query planner has statistics.
- Back up with `sqlite3 <file> ".backup <target>"`, not by copying the file while the site is busy.

Related pages

- Specifications: SQLite3 database driver, database drivers and installers, 16 to 30 September 2026, SQLite and Oracle driver behaviour (October 2026), platform SQLite installer
- SQLite: transactions queue for the write lock
- Install in one command, SQLite for Exponential Platform, MongoDB database support
- PHP 8 support (the countable and `is_countable` fixes that came out of the SQLite test runs)
- Changelogs: 6.0.1 (the driver), 6.0.13 (absolute paths, autoload), 6.0.14 (schema reading)
- History: January 2024, first half, January 2024, second half, April 2026, June 2026, second half (dropping indexes)

> **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.