Shared hosting

How to offer Exponential on shared hosting: what the sources say works there (Apache or LiteSpeed with .htaccess, the setup wizard, SQLite or the panel's database), what usually does not (Velocity), and the settings...

Source file: `doc/eng-us/help/hosting/models/shared-hosting.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/models/shared-hosting.md

This page is for hosting companies that want to offer Exponential on shared hosting: many customer accounts on one machine, managed through a panel, usually with FTP and without a shell for the customer. It answers what works there according to the project's sources, what usually does not, how a site gets installed without a shell, and what to configure per account.

> **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.

Contents

1. What the sources say
2. Serving: Apache or LiteSpeed with .htaccess
3. Installing without a shell
4. PHP settings per account
5. Databases on shared hosting
6. Cron jobs
7. Isolation between accounts
8. Checklist

What the sources say

- "Your hosting will not let you run a long-lived process. Shared hosting usually will not. Apache or LiteSpeed with `.htaccess` is the answer there." (root `README.md`, "When a

traditional server is still the right answer")

- Velocity "usually not" suits shared hosting because it "needs a long-running process"; Apache with PHP-FPM does (Chapter 8).
- "Shared hosting without shell access: the wizard. It needs only a browser and FTP access." (Chapter 4)
- cPanel and DirectAdmin hosts are tier 3: "shared hosting may forbid long-running processes, so Velocity may not be allowed; use Apache or LiteSpeed with `.htaccess`" (Tier 3).
- Since 6.0.15 the setup wizard "works with your own database only": it no longer runs `SHOW DATABASES` or opens the `mysql` system database, "which needs privileges a shared host does not give" (A clean installation that fits shared hosting).

Serving: Apache or LiteSpeed with .htaccess

The rewrite rules ship in the installation root as `.htaccess_root`. The customer (or your provisioning) copies it, so that an update can bring a new version:

Shell

```
cp .htaccess_root .htaccess
# or, where the account has a shell, as the account's user:
php bin/php/console exp:setup:htaccess --apply
```

The console command keeps the template's mode, refuses to overwrite a `.htaccess` that differs (it shows the diff; `--force` `--apply` replaces it after a backup to `var/backup/setup/<time>/`) and says what it would do when `--apply` is left out. `php bin/php/console exp:setup:config --apply` puts `config.php` into place the same way. Without a shell, the file manager or FTP client makes the copy. See Setting up .htaccess and config.php from the console.

Use `.htaccess_root_static` instead (`exp:setup:htaccess --static --apply`) only when the site serves a generated static cache. For Apache, the virtual host must let `.htaccess` work (`AllowOverride All` on the document root) and `mod_rewrite` must be on; "A 404 for every page means `.htaccess` is not read (`AllowOverride`) or `mod_rewrite` is off" (8.5 Apache with PHP-FPM). LiteSpeed "reads Apache's rewrite rules in most cases" and is tier 3 (Web servers).

> **Note**
>
> The document root of the account must be the installation root, the directory that holds `index.php`. "Safety comes from the routing, not from the directory layout: only a fixed list of asset paths is served as files; everything else is passed to `index.php`" (13.2). A panel's generic "serve the file if it exists" configuration would hand out `settings/*.ini` and the SQLite database; check once per server image that a request for `/settings/site.ini` returns an HTML page, never INI text.

More: Apache with PHP-FPM, LiteSpeed.

Installing without a shell

Without a shell the customer cannot run Composer, and "without Composer, or without running it, `vendor/` is missing and the site cannot start" (2.7 Composer). Two practical routes follow from that:

1. **The host provides the code.** Your provisioning (a one-click installer, a template account) places a tree in which `composer install` has already run, then the customer opens the setup wizard in the browser. See One-click installers.
2. **The customer prepares the code elsewhere.** The customer runs Composer on another machine with the same PHP version as the account and uploads the whole tree, `vendor/` included, by FTP.

Then the setup wizard does the rest in the browser: its **System check** page tests PHP, the extensions, the writable directories and the `php.ini` values, and prints the fix for each failure (Setup wizard on shared hosting, Chapter 5).

> **Warning**
> A long wizard step can hit the PHP-FPM request time limit of the account. The MongoDB porting notes describe the wizard being cut off by the pool's `request_terminate_timeout`, which PHP's own `set_time_limit()` cannot override (MongoDB kernel support). If the installs on your platform run long, raise the limit for the install and lower it again afterwards.

PHP settings per account

The setup wizard checks these (2.5 php.ini settings):

Setting	Value the wizard checks	Note for shared hosting
`memory_limit`	at least 64M	a floor; imports and image work need more
`max_execution_time`	at least 30	
`date.timezone`	set	
`file_uploads`	On	with a writable upload directory
`allow_url_fopen`	On	
`variables_order`	contains E	optional test
`open_basedir`	empty	optional test; it only warns: "open_basedir is in use and can give problems running Exponential due to bugs in some PHP versions."
`upload_max_filesize`, `post_max_size`	not tested	as large as the files editors upload

Panels often set `open_basedir` and `disable_functions` per account. The wizard only warns about `open_basedir`. The cronjob page in the administration needs `proc_open` to start a job, `exec` to read the crontab and `posix_kill` to stop a job; with one of them disabled "the page still opens and says which feature is missing" (Cronjobs, Before you start).

Details: PHP settings for Exponential, PHP selection and FPM pools.

Databases on shared hosting

In the official order: 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.

- **1. SQLite 3** needs no database server and no database account: the database is one file under `var/storage/sqlite3/`. On shared hosting that removes a whole provisioning step. It suits a site served from one machine, which a shared account is.
- **A database server from the panel** (commonly 4. MariaDB or 6. MySQL, sometimes 2. PostgreSQL): create the database and user before the install; "the installers do not create databases or users" (2.6 Databases). For MariaDB and MySQL, create the database in `utf8mb4`. The user needs only per-database privileges (9.3 MySQL and MariaDB).

See Databases and users in panels.

Cron jobs

Scheduled publishing, notifications and clean-up depend on `runcronjobs.php`. On shared hosting the customer usually schedules it through the panel's cron page. The command must use the same PHP version as the site; on machines with several PHP versions that means the full path to the right binary (Chapter 10, Cronjobs):

```
*/5 * * * * cd /var/www/vhosts/<domain>/<docroot> && /path/to/php runcronjobs.php -q -s site freq
uent >/dev/null 2>&1
```

See Cron and shell access.

Isolation between accounts

The README's warning applies to shared hosting too: one process that serves several sites can read every file the others hold. Give every account its own operating-system user and its own PHP-FPM pool running as that user, which is what Chapter 8 recommends when "several customers on one machine need operating-system isolation from each other". Inside the account, everything that writes `var/` (the pool, cron, the customer's scripts) should run as that one user (8.7.2 One user, or one group). See Multi-tenant isolation.

Checklist

- ☐ PHP 8.0 or later with the required extensions in the account's PHP version (Velocity runs on 8.0 too).
- ☐ Document root = installation root; `.htaccess` copied from `.htaccess_root`; `mod_rewrite` on.
- ☐ The `php.ini` values above set per account; `open_basedir` decision documented.
- ☐ `vendor/` present before the wizard starts.
- ☐ A database ready (SQLite needs nothing), created in UTF-8.
- ☐ A cron line for `runcronjobs.php` with the account's PHP.
- ☐ One user and one PHP-FPM pool per account.
- ☐ Backups of the database and of `var/storage`, `settings/override`, `settings/siteaccess`.

Related pages

- Choosing a hosting model
- Control panels: overview
- Setup wizard on shared hosting
- Setting up .htaccess and config.php from the console
- Chapter 8: Serving the site
- 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.