Configuring Exponential 6.0

Source file: `doc/eng-us/CONFIGURING.md` Last changed 2026-11-07
https://doc.exponential.earth/exphelp/viewer/CONFIGURING.md

This guide configures an Exponential 6.0 installation once it is installed: where settings live and which file wins, the siteaccesses and their addresses, languages, the database, mail, caches, file storage and sessions, security, the extensions, the cronjobs, debugging and logs, the help center and the website toolbar. It assumes a working installation from Installing Exponential 6.0 . Each section names the settings, shows a working example and links to the page with the full detail, so this page stays the map and the deeper pages stay the reference.

Every command is run from the installation root: the directory that holds `index.php`, `autoload.php`, `bin/` and `settings/`. The command console is `bin/php/console`; many installations add the shortcut `./console` (`ln -s bin/php/console console`). When you run a script as `root`, add `--allow-root-user`; the script says so when it is needed.

> **Never edit the defaults.** The files directly in `settings/` (`settings/site.ini`, `settings/file.ini` and the rest) and the `settings/` directories of the extensions are replaced by every update. Put every change of your own into `settings/override/` or `settings/siteaccess/<siteaccess>/`, as an `.ini.append.php` file. An update then keeps it, and the settings page and `exp:ini` show exactly where each value comes from.

Contents

1. The quick start: change a setting safely
2. Where settings live, and which file wins
3. Siteaccesses and their addresses
4. Languages
5. Database, mail, storage and sessions
6. Caches
7. Security
8. Extensions
9. Cronjobs
10. Velocity, Apache and PHP-FPM: when a change takes effect
11. Debugging and logs
12. The help center and the website toolbar
13. Troubleshooting

1. The quick start: change a setting safely

There are two ways to change a setting, and both write the same files.

In the administration: open **Setup > Ini settings** (`settings/view`), choose an INI file and a siteaccess, and click **Edit** beside the setting. **Save it in** offers *Siteaccess name only* (

`settings/siteaccess/<siteaccess>/`), *Every siteaccess (global override)* (`settings/override/`) or an extension. Saving clears the INI cache, so the next request uses the new value. Every setting shows the value in effect, the file it comes from and every other file that sets it. Passwords, keys and tokens are masked. The full walk-through: The settings page.

From a shell: `exp:ini` reads and writes the same scopes. It edits line by line, so the comments and the order of the file stay as you wrote them:

Shell

```
./console exp:ini where site.ini/SiteSettings/SiteName                 # which file wins, and th
e chain
./console exp:ini get site.ini/SiteSettings/SiteName global
./console exp:ini set site.ini/SiteSettings/SiteName "My site" global --dry-run
./console exp:ini set site.ini/SiteSettings/SiteName "My site" global
./console exp:ini toggle site.ini/DebugSettings/DebugOutput siteaccess:admin   # enabled <-> disa
bled
./console exp:ini scopes                                                # every place you can wr
ite to
```

The scopes are `global` (`settings/override/`), `siteaccess:<sa>`, `extension:<ext>` and `extension:<ext>:siteaccess:<sa>`. The defaults (`default`, `settings/<file>.ini`) are refused unless you add `--allow-default`. The reference: Change settings from the command line: exp:ini and console exp:ini.

With a text editor: an override file is PHP that hides the INI text in a comment, so a browser that requests it gets an empty page:

INI

```
<?php /* #?ini charset="utf-8"?

[SiteSettings]
SiteName=My site

*/ ?>
```

Then clear the INI cache: `php bin/php/ezcache.php --clear-tag=ini`. `config.php` switches off the check of file times (`EZP_INI_FILEMTIME_CHECK`), so a file changed in an editor is read again only after that clear. Under Velocity, some settings also need a restart (section 10).

2. Where settings live, and which file wins

Full guide: The settings page

Every INI file is read in layers. A later layer wins: it replaces a plain value, and adds to an array unless it empties the array first.

Order	Layer	Where	Shown on the settings page as
1	The defaults	`settings/<file>.ini`	Default
2	The extensions one siteaccess activates for itself (`ActiveAccessExtensions[]`)	`extension/<ext>/settings/<file>.ini.append.php`	Extension *name*
3	An extension's siteaccess	`extension/<ext>/settings/siteaccess/<siteaccess>/`	Extension *name* for *siteaccess*

Order	Layer	Where	Shown on the settings page as
4	The siteaccess	`settings/siteaccess/<siteaccess>/<file>.ini.append.php`	Siteaccess *name*
5	The active extensions (`ActiveExtensions[]`)	`extension/<ext>/settings/<file>.ini.append.php`	Extension *name*
6	The global override	`settings/override/<file>.ini.append.php`	Global override

This is the order of `eZINI::overrideDirsByScope()` (`lib/ezutils/classes/ezini.php`): the scopes `sa-extension`, `siteaccess`, `extension` and `override`, after the defaults. Layer 3 also holds an extension's `settings/override/` and `settings/override__<group>/` while one of its own siteaccesses runs (Per-site settings inside extensions); all of it stays below the siteaccess's own directory.

Inside one layer, the extension listed **first** wins: each extension is put in front of the ones before it, so the first one listed is read last. With `[ExtensionSettings] ExtensionOrdering=enabled` (the default) the list is first sorted by the dependencies in each `extension.xml`: an extension comes after the extensions it requires or uses, and before the ones it extends (The Extensions page). `./console exp:ini where <file>/<Block>/<Setting> <siteaccess>` lists the files of a setting in the order they are read.

What one line does to the value before it:

Line	What it does
`Name=value`	Sets a plain value; earlier values no longer count.
`Name[]`	Empties the array: what earlier files added is dropped.
`Name[]=value`	Adds an element at the end.
`Name[key]=value`	Sets the element *key*, replacing an earlier one with the same key.

So `settings/override/` wins over everything for every siteaccess. Use it for what is the same everywhere (the database, mail, the active extensions), and `settings/siteaccess/<siteaccess>/` for what belongs to one siteaccess (its design, its languages, its debug output).

An extension in `ActiveExtensions[]` still wins over the siteaccess: when it sets the same value, the line in `settings/siteaccess/<siteaccess>/` has no effect, and an array it fills cannot be emptied there. No directory that belongs to one siteaccess is read after the active extensions; an extension's own `settings/siteaccess/<siteaccess>/` is read *before* the siteaccess's directory, so it does not help. For a value of one siteaccess that must beat an extension:

- activate that extension in the `ActiveAccessExtensions[]` of the siteaccesses that need it instead of in `ActiveExtensions[]`: its settings are then read before the siteaccess's own;
- or, if the extension is yours, take the value out of its `settings/` and set it per siteaccess;
- or set it in `settings/override/`, when the same value may hold for every siteaccess.

Three more ways settings are layered, for larger setups:

- **Per environment:** with `EXP_ENV` in `config.env.php`, a variant file such as `settings/override/site.ini.test.append.php` is read beside the normal one, so a test copy can send mail to files without editing anything after a deployment. See Settings per environment.
- **Per site inside an extension:** an extension that holds one site of a multi-site installation can carry its own `settings/override/`, active only while one of its siteaccesses runs. See Per-site settings inside extensions.
- **Environment variables** for the values that differ per server or container: see Environment variables.

Every setting of every file is described with its default and an example under INI settings examples, for example site.ini [SiteSettings].

3. Siteaccesses and their addresses

Full detail: Siteaccess by host prefix, and the redirect to the language

A siteaccess is one face of the installation, with its own design, languages and settings. A new installation has three, always named `site` (visitors), `admin` (administrators) and `editor` (the administration for content only). What decides which one answers a request is `site.ini [SiteAccessSettings]`:

Setting	Meaning	Default
`AvailableSiteAccessList[]`	The siteaccesses that exist	set by the installer
`MatchOrder`	Which parts of the request are tried, in order: `uri`, `host`, `port`	`uri;host;port`
`URIMatchType`	How the path is matched: `element` takes the first path segment as the name, `map` uses `URIMatchMapItems[]`	`element`
`HostMatchType`	How the host is matched; `map` uses `HostMatchMapItems[]=host;siteaccess`	`map`
`[PortAccessSettings]`	A port per siteaccess, for example `8081=admin`	none
`[SiteSettings] DefaultAccess`	The siteaccess when nothing matches	`admin`

Three common layouts:

INI

```ini
# settings/override/site.ini.append.php: one host, the siteaccess as the first path segment
#   https://example.com/site/...   https://example.com/admin/...
[SiteAccessSettings]
MatchOrder=uri
URIMatchType=element
```

INI

```ini
# A host per siteaccess: https://www.example.com, https://admin.example.com, https://edit.example
.com
[SiteAccessSettings]
MatchOrder=host
HostMatchType=map
HostMatchMapItems[]
HostMatchMapItems[]=www.example.com;site
HostMatchMapItems[]=admin.example.com;admin
HostMatchMapItems[]=edit.example.com;editor
```

INI

```ini
# A port per siteaccess (development, or Velocity on several ports)
[SiteAccessSettings]
MatchOrder=port

[PortAccessSettings]
8080=site
8081=admin
8082=editor
```

Each siteaccess then sets its own address in `settings/siteaccess/<siteaccess>/site.ini.append.php`: `[SiteSettings]` `SiteURL` (the host and path that links and e-mail use, for example `www.example.com`), `SiteName`, and `DefaultPage`. A siteaccess becomes an administration through its design (`[DesignSettings]` `SiteDesign=admin4`) and the administration menu (`[AdminSiteAccessMenu]`). How the order of the siteaccess list is kept: The order of the siteaccesses. The editor siteaccess: The setup wizard and the editor siteaccess. A public siteaccess can let its visitors switch between designs with one click: Switching designs on a front siteaccess.

After changing the matching, clear the INI cache and check every address. If the administration no longer answers, read Troubleshooting.

4. Languages

Full guide: Content languages

The languages are set per siteaccess, in `site.ini` `[RegionalSettings]`:

Setting	Meaning
`Locale`	The interface language and the formats of dates and numbers, for example `ger-DE`
`ContentObjectLocale`	The language new content is created in
`SiteLanguageList[]`	The languages content is shown in, in order of preference: the first one that exists wins
`ShowUntranslatedObjects`	`enabled` shows content that has none of these languages, in its main language
`TranslationSA[]`	The siteaccess of each language, for the language switcher

INI

```ini
# settings/siteaccess/site_ger/site.ini.append.php: a German site that falls back to English
[RegionalSettings]
Locale=ger-DE
ContentObjectLocale=ger-DE
SiteLanguageList[]
SiteLanguageList[]=ger-DE
SiteLanguageList[]=eng-GB
```

A language is added to the installation once, under **Setup > Languages**. Its locale file in `share/locale/` gives its name, formats and text direction, and the administration has a full translation in many of them. The list of locales and how complete each translation is: Languages in Exponential. Languages under their own path (`/ger`, `/eng`) and the redirect from `/` to the visitor's language: Siteaccess by host prefix, and the redirect to the language. More: Translations and languages.

5. Database, mail, storage and sessions

Database

`site.ini` `[DatabaseSettings]`, in `settings/override/site.ini.append.php` (the installer writes it there):

INI

```ini
[DatabaseSettings]
# ezmysqli, ezpostgresql, sqlite3, mongodb, or ezoracle (extension ezoracle)
DatabaseImplementation=ezmysqli
Server=localhost
Port=
User=exponential
Password=
Database=exponential
Charset=utf-8
```

Keep the password out of version control: `settings/override/` is never committed. A changed engine needs the same data in the new database; see Upgrading and the database chapter of the installation book. SQLite under production load: Installing Exponential 6.0, section 9. Every setting: [DatabaseSettings].

Mail

`site.ini` `[MailSettings]` decides how mail leaves the server and who sends it:

Setting	Meaning
`Transport`	`sendmail` (default), `SMTP`, or `file` (writes every mail to `var/log/mail/` instead of sending it, for tests)
`TransportServer`, `TransportPort`, `TransportConnectionType`	The SMTP server; `tls` or `ssl` for an encrypted connection
`TransportUser`, `TransportPassword`, `TransportAuthMethod`	SMTP sign-in; empty method: the strongest the server offers
`AdminEmail`	The administrator's address, for system mail

Setting	Meaning
`EmailSender`	The `From` address of the mail the site sends; empty: `AdminEmail`

INI

```
# settings/override/site.ini.append.php
[MailSettings]
Transport=SMTP
TransportServer=smtp.example.com
TransportPort=587
TransportConnectionType=tls
TransportUser=site@example.com
TransportPassword=
AdminEmail=webmaster@example.com
EmailSender=no-reply@example.com
```

A test or staging copy should use `Transport=file`, best in an environment variant (section 2), so it never mails real people. More: [MailSettings], Mail transport hardening, Mail layout and Mail preferences.

File storage and clusters

Uploaded files and images go to `var/storage/` (`site.ini [FileSettings] VarDir=var`, `StorageDir=storage`), caches to `var/cache/`, logs to `var/log/`. A multi-site installation can give each site its own directory: Log and cache paths per site.

Several web servers that share one site share the files through the cluster file handler, `file.ini [ClusteringSettings] FileHandler`: `eZFSFileHandler` (the default, one server) or `eZDFSFileHandler` (a shared mount, `[eZDFSClusteringSettings] MountPointPath`, with the file metadata in the database through `DBBackend`). Every setting: [ClusteringSettings].

Sessions and "Remember me"

Full guide: Sessions

`site.ini [Session]`: `SessionTimeout` is how long a session lives after its last use, `RememberMeTimeout` how long the browser keeps the cookie when **Remember me** is checked. Both default to 31104000 seconds (360 days), and the box is checked on the login forms by default, because phone browsers drop session-only cookies on their own. Shorten them for a site with sensitive data:

INI

```
[Session]
SessionTimeout=1209600
RememberMeTimeout=1209600
```

Who is signed in, and signing browsers out: **Setup > Sessions** (Sessions); a user's own devices: Your devices and sessions.

6. Caches

Full guide: Caches

The caches are on by default and should stay on: `site.ini [TemplateSettings] TemplateCache=enabled` (compiled templates), `[ContentSettings] ViewCaching=enabled`

(rendered pages, cleared on publish). Clear the smallest one that fixes the problem:

You changed	Command
a setting (`.ini`)	`php bin/php/ezcache.php --clear-tag=ini`
a template (`.tpl`)	`php bin/php/ezcache.php --clear-tag=template`
content looks stale	`php bin/php/ezcache.php --clear-tag=content`
added a template or an extension	`php bin/php/ezcache.php --clear-id=template-override`
unsure	`php bin/php/ezcache.php --clear-all`

The same groups are in **Setup > Cache management**. Two more layers, both off by default:

- **The static cache** (`[ContentSettings] StaticCache=enabled`, `settings/staticcache.ini`) writes finished pages as files that the web server answers without PHP: The static cache generator. To publish a static copy to GitHub Pages or any git host: Static publishing.
- **Velocity's response cache** answers repeated requests from memory: Velocity response cache.

Warming the caches after a deployment: Preloading caches.

7. Security

Full guide: Security and audit

HTTPS and the security headers

Serve every siteaccess over HTTPS (Installing Exponential 6.0, section 8). `site.ini [HTTPHeaderSettings] SecurityHeaders[]` sends `X-Content-Type-Options`, `X-Frame-Options`, `Referrer-Policy`, `Permissions-Policy` and a frame-ancestors policy with every page. Once the whole site is HTTPS-only, switch on HSTS:

INI

```
# settings/override/site.ini.append.php
[HTTPHeaderSettings]
SecurityHeaders[Strict-Transport-Security]=max-age=31536000
```

The Content Security Policy

`site.ini [ContentSecurityPolicy] Mode` is `disabled`, `report-only`, `enforce` or `auto` (the default: report-only on the administration siteaccesses, no policy on the sites). `Directives[]` lists what a page may load. Watch the reports first, then enforce: Content Security Policy.

Behind a proxy or load balancer

Forwarded headers (`X-Forwarded-For`, `X-Forwarded-Proto` and the rest) are believed only from the addresses in `site.ini [HTTPHeaderSettings] TrustedProxies[]` (by default this machine only). Add your proxy's address or range; for the visitor's address also set `ClientIpByCustomHTTPHeader=X-Forwarded-For`:

INI

```ini
[HTTPHeaderSettings]
TrustedProxies[]
TrustedProxies[]=127.0.0.1
TrustedProxies[]=10.0.0.0/8
ClientIpByCustomHTTPHeader=X-Forwarded-For
```

See Trusted proxies.

Two-step sign-in and "Verify it's you"

- **Two-step sign-in** (the extension sevenx_authentication_2fa, activated by the installers) asks for a code from an authenticator app or by e-mail after the password, on the sign-in form. WebDAV, the REST API's basic authentication and SOAP ask for the password only, so keep them off or limited where the second step is required. Turning it on and requiring it: Two-step sign-in. Passkeys: Passkeys.
- **"Verify it's you"** asks a signed-in user to confirm again before a sensitive action (their own account, users and roles, system settings, exports, deletions). Five groups are on by default; the sixth, content, is switched on under **Setup > Verify it's you** (`/setup/reauth`). See Verify it's you.

Secrets

Passwords and keys that tools need (data table connections, import sources, remote servers) belong in the secret store, not in an INI file: **Setup > Secrets** (policy `secrets/manage`). A tool keeps only the name of a secret. A name `env:<VARIABLE>` reads an environment variable of the PHP process instead, which suits containers. The values are encrypted with a key that is not in the database (`EXP_SECRET_KEY` or `var/secrets/exp_secret.key`): back up the key with the database. See The secret store.

Roles and policies

Every user gets their rights from roles: a role is a list of policies (module and function, with limitations), given to a user or a user group, optionally limited to a subtree or a section. Start from the roles a new installation has (Anonymous, Editor, Administrator; a site package can add more, such as Member), copy one rather than changing it, and give the smallest set that works. Walk-through: Roles and policies. Every change is recorded in the audit trail.

8. Extensions

Full guide: The Extensions page

An extension is active when it is listed in `site.ini [ExtensionSettings] ActiveExtensions[]` (every siteaccess, in `settings/override/`) or `ActiveAccessExtensions[]` (one siteaccess, in `settings/siteaccess/<siteaccess>/`). The order of the list is the loading order, sorted by the dependencies in `extension.xml` while `ExtensionOrdering=enabled`: the extension listed first wins, for templates and for settings. The settings of `ActiveExtensions[]` also win over the siteaccess's own, those of `ActiveAccessExtensions[]` do not (section 2).

- **Setup > Extensions** (`/setup/extensions`) activates, deactivates and orders them, checks their dependencies and writes the list for you.

- From a shell: `./console exp:ini add site.ini/ExtensionSettings/ActiveExtensions[] myextension global`.
- After activating one, regenerate the autoloads (`php bin/php/ezpgenerateautoloads.php -e`) and clear the caches (or `./console exp:velocity deploy`).

An extension that ships `share/db_schema.dba` gets its tables when it is activated during the install. On an existing installation, the upgrade step `6.0.15-active-extension-tables` (`./console exp:upgrade`) creates the missing tables of every active extension; nothing is dropped. Which extensions a new 6.0.15 installation has and switches on, and where their tables come from: Extensions of a new installation. The loading order in detail: Extension loading order. Building your own: Extensions.

9. Cronjobs

Full guide: Cronjobs

The cronjobs publish scheduled content, send notifications and newsletters, and clean up. They are grouped into parts in `cronjob.ini` (`[CronjobSettings] Scripts[]` is the global part; `[CronjobPart-<name>]` the others). List them and add them to the crontab of the site user, with your public siteaccess:

Shell

```
php runcronjobs.php --list
```

```
*/5 * * * * cd /path/to/installation && php runcronjobs.php --siteaccess=site frequent >> var/log
/cron-frequent.log 2>&1
17 * * * *  cd /path/to/installation && php runcronjobs.php --siteaccess=site infrequent >> var/l
og/cron-infrequent.log 2>&1
```

Setup > Cronjobs runs a part, shows its output and its last run. Velocity's scheduler can run them without the system cron. Pausing a part: Pausing cronjobs; from the console: Cronjobs from the console.

10. Velocity, Apache and PHP-FPM: when a change takes effect

Full detail: The settings page, "When the change takes effect"

The settings are the same for both servers; what differs is when a change is used.

Change	Apache with PHP-FPM	Exponential Velocity
An INI setting, saved on the settings page or with `exp:ini`	the next request	the next request
An INI file changed in an editor or by `git pull`	after `--clear-tag=ini`	after `--clear-tag=ini`
`velocity.ini`, `site.ini [DatabaseSettings]`, `[ExtensionSettings]`, `[FileSettings] VarDir`, `file.ini [ClusteringSettings]`	after `--clear-tag=ini`	after `./console exp:velocity restart`
A PHP class, or an activated extension	reload PHP-FPM	restart Velocity

The settings that need a Velocity restart are listed in `site.ini [SettingsViewSettings]`

`RestartSettingList[]`; the settings page marks them **Velocity restart**. After a code or settings deployment, one command runs every step in the right order (autoloads, caches, the FPM reload, the Velocity restart): `./console exp:velocity deploy`. Velocity's own configuration: The Velocity web server; its HTTPS certificates: Velocity HTTPS certificates.

11. Debugging and logs

Full detail: The debug bar

Debug output is off by default and should stay off for visitors. Switch it on for your own address only:

INI

```ini
# settings/override/site.ini.append.php
[DebugSettings]
DebugOutput=enabled
DebugByIP=enabled
DebugIPList[]
DebugIPList[]=192.0.2.10
```

Or for one siteaccess while you work on it: `./console exp:ini toggle site.ini/DebugSettings/DebugOutput siteaccess:admin`. `[TemplateSettings] Debug=enabled` marks the templates that wrote the page with HTML comments (in mail too, so use it on a development copy only). Every debug setting: [DebugSettings].

The logs are in `var/log/` (with per-site log directories, `var/<site>/log/`): `error.log`, `warning.log`, `notice.log`, `debug.log`, plus the cron and mail logs. Read them in the administration under **Setup > Log files** (The log viewer); they are rotated when one outgrows `[FileSettings] LogFileMaxSize` (Log rotation). An error page names a reference: Finding an error by its reference.

12. The help center and the website toolbar

The help center (`exphelp`) shows this documentation in the administration, in the reader's language when a translation exists. Its settings are in `settings/exphelp.ini`: the languages of the documentation (`[DocLanguages]`), what the search weighs (`[SearchSettings]`), the order of the pages in the tree (`[TreeSettings] PageOrder[]`), and the landing pages per audience. The policy `exphelp/read` opens it. See The help system and Help content.

The website toolbar (`expwt`) is the sidebar for editing the site from its own pages: edit, create, move and publish where you read. It is set in `expwt.ini [ExpwtSettings]`:

Setting	Meaning	Shipped
`Enabled`	The sidebar on the pages of a siteaccess	`disabled`; the installers set `enabled`
`ShowInAdmin`	The sidebar in the administration siteaccesses too	`disabled`
`ShowToAnonymous`	Visitors who are not signed in: `disabled`, `enabled` or `policy`	`disabled`
`SiteAccessList[]`, `SiteAccessExcludeList[]`	Only these siteaccesses, and never these	all where enabled

Who sees it is decided by the policies `expwt/*`: a new installation gives them to the Administrator and Editor roles only, never to Anonymous. An installation that still uses the old ezwt toolbar moves over with `./console exp:wt:migrate-from-ezwt --apply`. See Extensions of a new installation, "ezwt and expwt".

Then continue with the guides:

- Operating a site: caches, cronjobs, backups, logs and repairs.
- Deploying: serve the site with Apache and PHP-FPM, Velocity or FrankenPHP, over HTTPS.
- Security and audit: check the hardening, roles and policies, the audit trail.
- Templates and design: find the template that wrote a page, override it, switch the admin design.
- Updating Exponential 6.0: bring the installation up to date, keeping these settings.
- Every guide: the learning path.

13. Troubleshooting

When the table does not help, Getting Support shows where to ask.

Symptom	Cause and fix
A changed setting has no effect	The INI cache still holds the old value: `php bin/php/ezcache.php --clear-tag=ini`. Check with `./console exp:ini where <file>/<block>/<setting>` which file wins: a later layer (often `settings/override/`) may set it again.
The settings page says **not in effect yet**	The file was changed outside the page and the cache was not cleared. Clear the INI cache (**Setup > Cache management**, INI caches).
A change works on port 443 but not under Velocity	The setting is read only when Velocity starts (**Velocity restart** badge): `./console exp:velocity restart`.
An array lost its default elements	The override starts with `Name[]`, which empties the array. Leave that line out to add to the defaults.
After changing `MatchOrder` the administration does not answer	No probe matches its address. Use the port or the old path, or fix `settings/override/site.ini.append.php` in a shell and clear the INI cache.
Links and e-mail point to the wrong host	`[SiteSettings] SiteURL` of that siteaccess is still the installer's value.
Mail is not sent	`Transport=file` is set in an override or environment variant (mail goes to `var/log/mail/`), or the SMTP server refuses the sign-in: read `var/log/error.log`.
An activated extension does nothing	Autoloads not regenerated, or its tables missing: `php bin/php/ezpgenerateautoloads.php -e`, then `./console exp:upgrade` for the tables, then clear the caches.
Behind a proxy every visitor has the proxy's address	The proxy is not in `TrustedProxies[]`, or `ClientIpByCustomHTTPHeader` is not set.
The browser blocks scripts after switching the CSP to `enforce`	A directive is missing; go back to `report-only` and read the reports first.

Symptom	Cause and fix
A secret "cannot be decrypted here"	This server has another key: copy `var/secrets/exp_secret.key` or set EXP_SECRET_KEY to the key the secrets were written with.

More: Repairing an installation, Operating a site.

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.