Features

Features.

Last changed 2026-11-04
/exphelp/manual/help-gen-manual-features

4.7

5.2

5.4

Exponential 6.0 features

This page lists every feature page of the Exponential 6.0 line, so you can find the one you need without searching.

"This form has expired": form token refusals

This page is for site owners who see "form expired" pages or warnings, and for developers who handle them.

This page is for site owners who see "form expired" pages or warnings, and for developers who handle them. Exponential protects every form against cross-site request forgery with a form token (the `ezformtoken` extension in the kernel). When a POST arrives without its token, or with a wrong one, the visitor used to see a 500 page, and the log recorded "Unexpected error" for what is really a refusal. Since 27 September 2026 the answer is a clear page with the right status.

What a visitor sees

A page that says, in plain words:

- the form has expired: the page was open for a long time, or the form was sent from another page;
- **nothing was saved**;
- **Reload the form and send it again**, or go to the front page (the dashboard in the administration).

A signed-out visitor is told they may have to sign in again; when the refused form was itself a sign-in form, the page asks to sign in again. The wording is the same for a missing and a wrong token, and does not talk about security tokens and sessions.

The page is `design:error/kernel/6.tpl`, shipped in `design/standard`; the admin designs fall back to it. Check:

Shell

```
ls design/*/templates/error/kernel/6.tpl
```

While a design has no such template, or it renders empty because the template cache was built before it existed, a built-in explanation is shown.

What the browser and scripts get

	Value
HTTP status	`403 Forbidden` (kernel error 6, mapped in `error.ini`)
Headers	`Cache-Control: no-store`, the configured security headers
HTML request	Kernel error 6 through the error module, inside the pagelayout of the module that was posted to; no redirect; session and posted data are left as they are
XHR, `Accept: application/json`	A JSON body with the same message
Error view title and path	"Form expired" instead of "kernel (6)"

In the log, a refusal writes one line to `var/log/warning.log` with the check (missing or wrong), the method, the path and the siteaccess, **never a token**. The same refusal again (same check, siteaccess, path and client) within `RefusalLogCollapseSeconds` is counted instead of written, and the next line gives the count.

Settings

File	Block	Key	Default	Meaning
`settings/site.ini`	`HTMLForms`	`RefusalLogCollapseSeconds`	`60`	Collapse window for repeated refusals. `0` writes every one
`settings/site.ini`	`HTMLForms`	`Secret`	placeholder text	CSRF secret. Set your own per project
`settings/error.ini`	`ErrorSettings-kernel`	`HTTPError[6]`	`403`	Status of the refusal

For developers

- `ezpFormTokenException` carries the failed check and code 403. The old code threw a plain `Exception` that nothing caught until the uncaught handler.
- `ezpFormTokenRefusal` builds the answer: status, headers, the template parameters of `design:error/kernel/6.tpl` (`reason`, `reason_code`, `referrer`, `retry_url`, `is_ajax`, `title`, `message`, `action`), the JSON body and the built-in page.
- The web kernel catches the exception at `request/input`. A refusal escaping anywhere else (the REST or tree menu kernel, a module running the check itself, the uncaught exception handler) gets the same 403 and warning line.
- The built-in strings are literal `ezpI18n` calls in the contexts `kernel/error/formtoken` and `design/standard/error/formtoken`, translated into German and `eng-US`. An unmatched refusal no longer writes an `http_response_code() has no effect` warning.
- A page carrying a form token is marked `Cache-Control: private, no-cache, must-revalidate`, so no shared cache can store it with a token and hand it to an anonymous visitor.

Related pages

- Security defaults of September 2026
- Datatype and input hardening

- RAD security
- Behaviour changes, 16 to 30 September 2026
- Changelog 6.0.15
- Chronicle, 16 to 30 September 2026

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. Start with Partners: start here and How to apply.

Changing your password

user/password is the page where a signed-in user changes their own password, in the administration (/admin/user/password) and on the public site (/user/password).

`user/password` is the page where a signed-in user changes their own password, in the administration (`/admin/user/password`) and on the public site (`/user/password`). Since 6.0.15 the page lists the password's requirements beside the field, reports every problem next to the field it belongs to, and says clearly when the change has gone through. After a change the user gets an e-mail saying so. A site can add its own password rules.

In short

- The form still has three fields: current password, new password, and the new password again. Their names, the buttons and the redirects are the same as before, so forms in other designs and extensions keep working.
- The requirements are listed under the new password. The server checks them in every case. With JavaScript, the page also ticks them off while you type, shows a strength meter, tells you whether the two new passwords match, lets you show or hide each field, and can generate a strong password.
- A failed change names each problem next to its field and in a summary at the top. Each summary entry links to its field.
- The page never writes a typed password back into the HTML. Before 6.0.15 it put the typed passwords into the fields' `value` attributes.
- After a change you stay signed in, and your session gets a new id. Your other sessions (other browsers, other devices) are signed out on their next request, whatever session handler the site uses. This also happens when the password is changed anywhere else: by an administrator, in content/edit, through "Forgot your password?" or on the console.
- The same rules apply to every form that sets a password: registration, editing a user account in content/edit or in the administration, and the console command `bin/php/resetuserpassword.php`.
- An e-mail, "Your password was changed", goes to the account's address. It never contains the password. It belongs to the essential category **Account security**, so nobody can switch it off.
- Every change and every refused change is recorded in the audit trail. The password is never recorded.

The rules

The length rule has not changed: `site.ini [UserSettings] MinPasswordLength` (10 on a new installation; 3 when the setting is missing). Length is counted in bytes, as before, so no password that passed before is refused now.

The other rules are in `site.ini [PasswordSettings]`. They are all **off** by default, so an upgrade does not make the rules any stricter:

Setting	Rule id	The new password must
`RequireLowercase=enabled`	`lowercase`	contain a lowercase letter, in any script
`RequireUppercase=enabled`	`uppercase`	contain an uppercase letter
`RequireDigit=enabled`	`digit`	contain a digit
`RequireSymbol=enabled`	`symbol`	contain a character that is neither a letter nor a digit (a space counts)
`MinCharacterClasses=N`	`classes`	mix at least N of those four kinds (0 to 4; 0 = off)
`ForbidLogin=enabled`	`not_login`	not contain the login name (case is ignored; only logins of 3 or more characters)
`ForbidCurrentPassword=enabled`	`not_current`	differ from the current password

Since 6.0.15 one more rule is **on** by default, in `site.ini [UserSettings]`:

Setting	Rule id	The new password must
`WeakPasswordCheck=enabled`	`not_common`	not be a well-known or common password, nor a simple variant of one (`expWeakPasswordList`; more words with `WeakPasswordList[]` and `WeakPasswordFile`)

It applies to new passwords only; nobody is refused at sign-in. `WeakPasswordCheck=disabled` brings back the rules of before. Details: doc/eng-us/bc/6.0/weak-passwords-and-reset-key-lifetime.md.

To turn a rule on, use an override, for example `settings/override/site.ini.append.php`:

INI

```ini
[PasswordSettings]
MinCharacterClasses=3
ForbidLogin=enabled
```

The rules live in one class, `expPasswordPolicy` (`kernel/classes/exppasswordpolicy.php`). Its `rules()` method returns the list, `validate( $password, $user )` returns the ids of the rules a password fails, and `errorText( $id )` returns the error sentence. `accountErrors( $password, $user, $login )` returns the sentences for a user account form; the user account datatype (registration, content/edit, the administration's user edit) and `bin/php/resetuserpassword.php` use it, so every form refuses the same passwords. The length sentence there is the one of before; an unchanged password on the edit form (`_ezpassword`) is not checked.

After a change

Setting (`[PasswordSettings]`)	Default	What it does
`EndOtherSessions`	enabled	gives the current session a new id and ends the user's other sessions (also switches the password stamp, below, on and off)
`ChangeNotificationMail`	enabled	sends "Your password was changed" to the account's address
`StrengthMeter`	enabled	tells the page's script to show the strength meter

Setting ([PasswordSettings])	Default	What it does
GenerateButton	enabled	tells the page's script to offer "Generate a strong password" ([UserSettings] GeneratePasswordLength characters, at least 12)

Other sessions: the password stamp

At sign-in, the session stores a **password stamp** in eZUserPasswordStamp, next to eZUserLoggedInID. The stamp is the user id plus a keyed hash (HMAC-SHA256, with a key derived from the site secret) of the user id and the stored password hash. It never contains the password or its hash, and it changes whenever the password changes.

On every request, eZUser::currentUser() compares the session's stamp with the stamp of the user's current password. It costs no extra database query: the hash comes from the user cache that currentUser() reads anyway, and the comparison runs in constant time (hash_equals). When the stamps differ, the session is signed out (eZUser::logoutCurrent()) and access.session.revoke is recorded with reason password_changed.

- **The session that made the change stays signed in.** eZUser::store() gives the session of the user who was stored the new stamp, and refreshes the user object that the request keeps.
- **Sessions from before the upgrade stay valid.** A session without a stamp is not checked; it gets a stamp at its next sign-in. A deploy does not sign anybody out.
- **Not checked:** the anonymous user; accounts without a password of their own (an empty hash, as for LDAP, text-file and single sign-on users); a temporary switch to another user (eZUser::NO_SESSION_REGENERATE, used by the preview caches); and every session when EndOtherSessions=disabled.
- The user cache is a PHP file that is read with include, so a server's OPcache can keep an outdated copy until it revalidates. When a stamp does not match, the check reads the stored row once, which costs one query and only in that case, so an outdated cache never signs out the session that signed in with the new password. eZUser::purgeUserCacheByUserId() also invalidates the file in OPcache. A change made through the site therefore reaches every worker of that server at once. A change made on the console (resetuserpassword.php, a script) reaches a web server's other sessions when that server's OPcache revalidates (opcache.revalidate_freq; 10 seconds on the demonstration server's Apache, at once on Velocity).
- When [UserSettings] UpdateHash re-hashes a password at sign-in (after a change of HashType), the user's other sessions are signed out once, because the stored hash has changed.

With the database session handler ([Session] Handler=ezpSessionHandlerDB), the other sessions are also deleted at once, at the time of the change (ezpSessionHandler::deleteOtherSessionsOfUser()). A handler in an extension can support this by overriding that method.

The mail

The mail uses the template design:user/password_changed_mail.tpl, which a design can override. Its variables are $user, $changed_at (a timestamp), $ip, $sessions_ended, $other_sessions_signed_out and $site_name, and the template may set $subject. The mail goes through the mail gate in the category security. A user without a valid address gets no mail.

The audit trail

Event	When
`access.user.password.change`	a password is changed (unchanged; recorded by `eZUser::store()`)
`access.user.password.change.failed`	a change is refused; `after.rule` is `old_password`, `confirmation`, or the id of the first rule that failed
`access.session.revoke` (new)	the user's other sessions were ended: deleted at once with the database handler (reason `password_change`, `after.count` the number), or one session signed out on its next request by the password stamp (reason `password_changed`)

For template authors

The page gets new template variables, all optional: `$password_changed`, `$password_rules`, `$field_errors`, `$error_summary`, `$field_ids`, `$min_length`, `$password_js_config_json` and a few more. Old templates keep working unchanged. The variables and the `data-exp-password-*` hooks of the shared script are listed in the compatibility note.

Demo mode

On a public demo (environment variable `EXP_DEMO_MODE=1`) the page refuses to change the password of an administrator: it shows "This installation is a public demo: the passwords of administrators cannot be changed or reset." before anything is typed (`demo_mode_locked`, `demo_mode_text`) and in the error summary after a post, and nothing is changed. A design that overrides `user/password.tpl` shows the notice with `first_set( $demo_mode_locked, false() )`. Demo mode.

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. Start with Partners: start here and How to apply.

E-mail preferences: the user's guide

Exponential lets every person choose which e-mail they get from a site, on one page, and stop it at any time with one click.

Exponential lets every person choose which e-mail they get from a site, on one page, and stop it at any time with one click. This page is for visitors, members and editors: where the page is, what each choice means, how to unsubscribe from a mail, what happens when you sign up, and how to download your data. Administrators find the setup, the status page and the suppression list in the administrator's guide; developers find categories and the mail gate in the developer's guide.

In short

- Optional e-mail (notifications, newsletters, offers, notices) is **off until you turn it on**.
- Essential e-mail about your account (password reset, activation, orders, legal notices) is always sent and never carries advertising.
- Every optional e-mail ends with why you got it, an **Unsubscribe** link that works with one click, and a link to your preference page. No login is needed for either.
- Your choices, and every change to them, can be downloaded as JSON or CSV.

Where the page is

You are	Open
Signed in on the public site	your profile (**Edit profile**), the box **E-mail preferences** > **Open my e-mail preferences**, or `/mailpreferences/settings`
Signed in to the administration (admin, editor)	**My account** > **My e-mail preferences**, the link in the current user box, or the box at the top of **My notification settings**
Not signed in, or without an account	the **Manage** link at the end of any optional e-mail, or `/mailpreferences/request` ("Send me a link")

The page looks the same everywhere. It has four parts, top to bottom.

1. Send me optional e-mail (the main switch)

One line says whether optional e-mail is **On** or **Off**, with one button:

- **Turn off all optional e-mail** stops every optional kind at once. Your choices below are kept, dormant.
- **Turn on optional e-mail** brings them back exactly as they were.

Essential e-mail is not affected by this switch.

2. What you want to receive

One switch per kind of e-mail, with its name and what it contains. Nothing is on until you turn it on. Tick what you want and press **Save my choices**; untick and save to stop it. The notice at the top says what happened ("Your choices were saved.", "Nothing was changed.").

Kind (default installation)	What it is	How often
Content notifications	New and changed content in the parts of the site you follow	At once, Daily summary or Weekly summary
Collaboration and approvals	Content waiting for your approval, messages of your collaboration items	At once, Daily summary or Weekly summary
Newsletters	The newsletters you subscribe to	the newsletter's own schedule
Offers and news	Offers, events and news about products and services	the sender's schedule
Application and system notices	Notices about the site and its features that are not about your account	when it happens

A site can add kinds of its own; they appear in the same list.

Newsletters and **Offers and news** need a confirmation: when you turn one on, the page says "We sent a confirmation link to ... This starts once you open it." and the switch shows **Waiting for your confirmation**. Open the link in that mail and press **Yes, confirm**. Until you do, nothing of that kind is sent; after seven days the request lapses and the switch is off again. Content notifications and the others start at once.

If you followed items before this page existed (**My notification settings**), Content notifications already shows as on, with the frequency of your digest. What you choose here wins.

3. Always sent

The essential kinds, listed with their description: Account security, Orders and receipts, Legal and service notices, Administration alerts. "These messages are needed to run your account or are required by law. They cannot be turned off, but they never contain advertising."

4. Your e-mail data

The last 20 changes to your preferences: when, what, the change, where it was made (Preference page, Link in an e-mail, Registration form, Confirmation link, Changed by an administrator, ...) and the exact text you saw when you made it. **Download my e-mail data (JSON)** and **(CSV)** give you everything: each kind with its state, pending confirmations and the whole history.

Unsubscribing from an e-mail

Every optional e-mail ends like this (text version; the HTML version has links):

```
--
You get this e-mail because you turned on "Content notifications" on Example Site.
Unsubscribe with one click: https://www.example.com/mailpreferences/unsubscribe/m1...
Choose which e-mail you get: https://www.example.com/mailpreferences/manage/m1...

Example Organisation
Example Street 1, 12345 Town
```

- **Unsubscribe** opens a page that asks "Stop e-mail of the kind "Content notifications"?" with one button, **Unsubscribe**. Pressing it stops that kind at once: "... gets no more e-mail of the kind ...". Opening the link alone changes nothing, so a virus scanner that opens links cannot unsubscribe you.
- Many mail programs (Gmail, Apple Mail, Outlook, Yahoo) show their own **Unsubscribe** button next to the sender. It uses the same link and unsubscribes you without opening a page.
- **Choose which e-mail you get** opens your full preference page, without login.

The links work as long as the site exists; they are personal, so do not forward a mail with them.

A link that does not work any more (cut off by the mail program, for example) shows the "Send me a link" form with the notice "This link does not work any more".

Send me a link (without an account)

`/mailpreferences/request`: type your address and press **Send me the link**. The page answers "Please check your inbox. If we send e-mail to ..., a message with your personal link is on its way. The link works for 24 hours." It says the same for every address, so nobody can find out from it who is on a list. One address can ask for three links an hour.

When you register

The registration form has a box **E-mail from us (optional)** with one unticked box per optional kind. Tick only what you want; nothing is ticked for you. A kind that needs a confirmation sends its link after the registration. What you ticked is recorded with the text the form showed.

Changing your e-mail address

A new address must be confirmed: a link goes to the new address ("Use ... for this account?"), and the account keeps the old address until you press **Yes, confirm**.

Your rights

- **See your data**: the history on the page and the download.
- **Withdraw**: any switch, the main switch, or the unsubscribe link, at any time.
- **Erase**: ask the site's administrator. Your preferences are removed and the history is anonymised; a

record that you withdrew is kept without your address, so the site can prove it stopped writing to you.

- **Stop everything for good**: the unsubscribe link of a mail sent to everyone ("Stop all optional e-mail?") also puts your address on the site's block list (only a scrambled form of the address is kept). Turning optional e-mail on again on your page lifts that block.

Languages

The pages and mails are in English and German; the German site (for example `/bold_ger/mailpreferences/settings` on a demo installation) shows "E-Mail-Einstellungen".

Related pages

- E-mail preferences: the administrator's guide
- Notifications: the user's guide
- The law checklist

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. Start with Partners: start here and How to apply.

Reset a user password from the command line

This page is for administrators who have lost the administrator password, have taken over an installation, or have imported users. bin/php/resetuserpassword.php sets a new password for any user.

This page is for administrators who have lost the administrator password, have taken over an installation, or have imported users. `bin/php/resetuserpassword.php` sets a new password for any user. It was added on 11 July 2026 (`ea77927c23`).

The new password is hashed by the same code as the user edit form (`eZUser::setInformation`), and it must satisfy `site.ini [UserSettings] MinPasswordLength`.

Reset the admin password now

As the operating system root user, generate a new 20-character password for `admin`:

Shell

```
php bin/php/resetuserpassword.php --allow-root-user -u admin -g -l 20
```

The generated password is printed once. Note it, sign in, and change it in the admin if you like.

Two ways to authorise

As the operating system root user (no Exponential login needed):

Shell

```
php bin/php/resetuserpassword.php --allow-root-user -u admin -p 'NEW_PASSWORD'
```

`--allow-root-user` (short `-r`) requires that you really are the OS user `root`; otherwise the script stops with an error.

As an administrator of the site:

Shell

```
php bin/php/resetuserpassword.php -a admin -ap 'YOUR_ADMIN_PASSWORD' -u editor1 -p 'NEW_PASSWORD'
```

The login given with `-a` and `-ap` must authenticate and must hold the `Administrator` role.

Options

Option	Meaning	Default
`-a <login>`	Administrator login that authorises the reset	none
`-ap <password>`	Password of that administrator	none
`-r`, `--allow-root-user`	Skip administrator authentication; OS root only	off
`-u <login>`	Login of the user to change	`admin`
`-p <password>`	The new password	none
`-g`, `--generate`	Generate a random password instead of `-p`	off
`-l <length>`	Length of a generated password	`16`
`-s`, `--siteaccess`	Siteaccess (and so database) to use	default access
`-q`, `--quiet`	Only errors are printed	off
`-h`, `--help`	Help	

Errors you may meet

Message	Cause
`Admin authentication required`	Neither `-a`/`-ap` nor `--allow-root-user` was given.
`Admin authentication failed.`	Wrong login or password after `-a`.
`... does not have the Administrator role.`	The authorising user is not an administrator.
`Target user not found: <login>`	`-u` names a login that does not exist.
`Target password does not validate. It must be at least N characters long.`	Shorter than `MinPasswordLength`.

Good practice

A password given on the command line is visible in the process list and in your shell history. Prefer `-g` and change the password in the admin after the first sign-in, or clear the history afterwards.

Demo mode. With the environment variable `EXP_DEMO_MODE=1` (a public demo) the command refuses to reset the password of an administrator, also with `--allow-root-user`: it prints "This installation is a public demo: the passwords of administrators cannot be changed or reset." and exits with status 1. Other accounts are reset as usual. A demo does not need it after the installation: `exp:install` installs the published password as given in demo mode. Demo mode.

Related pages

- The August 2026 security patches (sign-in checks)
- Security defaults of September 2026

- Exponential Console
- Audit trail
- Chronicle: July 2026

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. Start with Partners: start here and How to apply.

Extensions, themes and packages

This page lists every legacy extension, theme and package of Exponential 6 that has its own page.

Features of the 5.90 series (2017.08 and later)

This folder contains features added to "5.90" release, an alias for 2017.08 and higher releases.

GNU Free Documentation License 1.3