Middleware: request rules, input and output listeners

What takes the place of middleware, before and after filters and request listeners in Exponential - request rules that allow, deny, redirect, rewrite or log after URL translation and policies, the request/input and...

Source file: `doc/eng-us/help/web-framework-developers/middleware-and-request-rules.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/middleware-and-request-rules.md

This page is for developers who put cross-cutting request logic into middleware (Laravel, Slim, Laminas, Express, Django), before and after filters (Rails, CodeIgniter) or kernel listeners (Symfony). Exponential has no generic middleware stack you add classes to. It has three places that together cover most middleware uses.

Contents

1. The three places
2. Request rules
3. Input and output listeners
4. Common middleware, translated
5. What has no equivalent

The three places

Place	Runs	Can	Configured in
`request/preinput`, `request/input` listeners	before URL translation and routing	inspect the request, refuse it (as the CSRF check does), set up state	`site.ini [Event] Listeners[]`
Request rules	after URL alias translation and after the policies allowed the view	allow, not found, forbidden, login, redirect, redirect to alias, rewrite, log	`settings/requestrules.ini`
`response/preoutput`, `response/output` filters	after the pagelayout rendered	change the final output (as the form token insertion does)	`site.ini [Event] Listeners[]`

The order inside a request is on The request lifecycle.

Request rules

Rules are settings: conditions and an action, asked in order, the first whose conditions all match decides (Request rules, complete guide with sixteen recipes: View full security).

INI

```ini
# settings/siteaccess/<site>/requestrules.ini.append.php
<?php /*
[RequestRuleSettings]
RuleList[]=members_area

[Rule-members_area]
Description=Below node 864, anonymous visitors sign in first and come back.
Conditions[subtree]=864
Conditions[user]=anonymous
Action=login
*/ ?>
```

- 25 conditions, among them `requested_via` (system, alias, wildcard, front page), `uri`, `module_view`, `view_mode`, `node`, `subtree`, `class`, `section`, `state`, `user`, `role`, `policy`, `siteaccess`, `host`, `header`, `ip` and `fact`.
- 8 actions: `allow`, `notfound`, `forbidden`, `login`, `redirect`, `redirect_to_alias`, `rewrite`, `log`.
- Rules only take access away or send the visitor elsewhere; they never give access a policy refused.
- Any answer except `allow` is sent `private, no-store`.
- Check what the rules decide without changing anything: `php bin/php/ezrequestrules.php -s site --uri=content/view/full/2`.

In PHP you can add conditions, actions, facts (`ezpRequestFactProvider`) and whole rule sets computed at run time (`ezpRequestRuleProvider`), registered in `[RequestRuleSettings]` (Extending the rules in PHP).

Input and output listeners

`ezformtoken` is the model of a middleware pair in Exponential. Its `settings/site.ini.append.php` registers:

INI

```ini
[Event]
Listeners[]=request/input@ezxFormToken::input
Listeners[]=response/output@ezxFormToken::output
```

`ezxFormToken::input()` checks the token of every POST before routing; `ezxFormToken::output()` adds the hidden token field to every `method="post"` form in the final HTML. A filter listener such as `response/output` must return the value it receives, changed or not (Events and listeners).

Common middleware, translated

Middleware	In Exponential
Authentication required for an area	siteaccess `RequireUserLogin`; a request rule with `Action=login`; view `functions`
Authorization	policies on the view (`functions`), object checks in the view (Authorization)
CSRF	`ezformtoken` (Forms and CSRF)

Middleware	In Exponential
IP allow list for an area	a request rule with `Conditions[ip]`
Redirect old addresses	URL wildcards or a request rule with `redirect` (Pretty URLs)
Maintenance mode	`expMaintenance` in `index.php` (`php bin/php/maintenance.php on`) (Maintenance mode)
Security headers, Content Security Policy	`site.ini [ContentSecurityPolicy]` and the security headers (Content Security Policy)
Response caching	the view cache, the role-aware HTTP cache, the Velocity response cache (Caching)
Request logging	request logs and the audit trail (Request logs, Audit trail)
Trusted proxies	`site.ini [HTTPHeaderSettings] TrustedProxies[]` (Trusted proxies)

What has no equivalent

- **Per-route middleware lists** (`->middleware('auth')` on one route): use view `functions` and request rules with a `module_view` condition.
- **Wrapping the whole call in a closure** (`$next($request)`): listeners run before or after, not around. The runnable events (`runnable/view/before` and `after`) come closest for a single view.
- **Middleware ordering by priority**: listeners run in the order of `Listeners[]`; rules in the order of `RuleList[]`.

Related pages

- Hardening
- Security for applications
- Event hook pattern
- 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.