Monitoring integrations

Knowing that an integration works before the customer notices that it does not - what to watch (runs, freshness, backlog, errors, credentials), the signals Exponential provides (summary lines, cronjob pages, the audit...

Source file: `doc/eng-us/help/integrators/monitoring-integrations.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/integrators/monitoring-integrations.md

This page is for operators and integrators who keep integrations running. An integration usually fails silently: the cronjob stops, a key expires, the other system changes a field, and the site simply shows old data. Monitoring turns those silent failures into alerts. The server side of monitoring is in Monitoring overview and Health checks and uptime.

Contents

1. What to watch
2. Signals Exponential provides
3. Checks to set up
4. Alerts that help
5. A monitoring plan

What to watch

Signal	Question	Example threshold
Run happened	Did the nightly import run?	Alert when no summary line for 26 hours
Run succeeded	Did it finish without failures?	Alert on `failed > 0` or a non-zero exit code
Freshness	Is the newest imported change recent?	Alert when the newest `modified` of imported objects is older than 2 days
Backlog	Is the outbox growing?	Alert when more than 500 unsent rows or the oldest unsent is older than 1 hour
Dead letters	Did anything give up?	Alert on any
Errors	New entries in `error.log` naming your extension	Alert on a rate, not each line
Credentials	Will a key expire soon?	Warn 14 days before; Setup > oAuth admin > API keys filters "expiring within seven days"
Volume	Is today's number of records plausible?	Warn when it differs from the average by more than a set share

Signals Exponential provides

Signal	Where
Your summary lines	`var/log/<your log>.log` (`eZLog::write()`), searchable with `php bin/php/console exp:logs --tail=... --search=...`
Cronjob runs and output	**Setup > Cronjobs**: last runs, issues, the output of a running part (Cronjobs)
Errors with a reference	`error.log`, the log viewer at **Setup > Log files**, `exp:logs --reference=ERR-...`
Workflow processes waiting	**Setup > Workflow processes**; a growing list means the workflow cronjob is not running
API key use and refusals	Audit events `access.apikey.use.failed` with reasons; `php bin/php/console exp:audit tail --channel=access`
Audit sinks	`php bin/php/console exp:audit sinks list`: ready or not, how much is spooled
Job queue	**Setup > Job queue** and `php bin/php/console ext:expqueue:status` when expqueue is used
Installation health	**Setup > System information** and `php bin/php/systeminfo.php --json` (System information)
Maintenance mode	`php bin/php/maintenance.php status`

Checks to set up

A small check script of your extension, run by the monitoring system (or by cron with its output mailed to the operators), turns these signals into a status:

PHP

```php
<?php
// extension/myintegration/bin/php/check.php: exit 0 OK, 1 WARNING, 2 CRITICAL; one line of outpu
t
// (bootstrap with eZScript as in the CLI chapter, then:)
$db = eZDB::instance();
$pending = $db->arrayQuery( 'SELECT COUNT(*) AS n, MIN(created) AS oldest FROM myintegration_outb
ox WHERE sent = 0' );
$n = (int)$pending[0]['n'];
$age = $pending[0]['oldest'] ? time() - (int)$pending[0]['oldest'] : 0;
$status = ( $n > 500 || $age > 3600 ) ? 2 : ( $n > 100 ? 1 : 0 );
$cli->output( array( 'OK', 'WARNING', 'CRITICAL' )[$status] . " outbox pending=$n oldest_age={$ag
e}s" );
$script->shutdown( $status );
```

Expose the same check through a protected endpoint only if the monitoring system cannot run commands on the server.

Alerts that help

- Say **what** stopped and **since when**, and link the runbook section ("Product import: no run since 02:15; see runbook 3.2").
- Alert the people who can act: the operator for a stopped cronjob, the customer's ERP team for a broken source file.

- Avoid alert storms: one alert per condition, repeated at a slow rhythm, cleared automatically when the condition ends.
- Keep warnings out of the night; page only for what cannot wait until morning.

A monitoring plan

```
Integration: ERP product import                          Owner: ops@example.com
Check                  How                                  Warning        Critical       Who
Run happened           summary line in myintegration.log    > 25 h         > 26 h         oper
ator
Run succeeded          exit code / failed=               skipped > 50   failed > 0     operato
r
Freshness              newest modified of erp-product-*     > 1 day        > 2 days       cust
omer ERP team
Outbox backlog         check.php                            > 100          > 500 / 1 h    oper
ator
Key expiry             API keys page                        < 14 days      < 3 days       oper
ator
Error rate             error.log lines naming myintegration > 10 / h       > 100 / h      deve
loper
```

Related pages

- Error handling and logging
- Monitoring overview for hosts
- KPIs from logs and the audit
- Handover of an integration
- 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.