Request logs

How Exponential separates real faults (error.log) from not-found answers (404.log) and suspicious requests (security.log), what each line holds, how addresses are masked, which settings control it and how support staff...

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

This page is for system administrators and support staff of a host. It answers: where a request for a missing page and a request that looks like an attack are logged, what a line tells you, how personal data is kept out of the files, how much disk they take, and how to use them when a customer asks "why are visitors getting errors?".

Every format and setting below is from Request logs: 404.log and security.log, which also covers the developer interfaces. The security side has its own page: Security log.

Contents

1. Three logs instead of one
2. Reading a 404.log line
3. Privacy by default
4. Settings
5. Disk use and rotation
6. Using the logs in support
7. Checklist

Three logs instead of one

Exponential writes a not-found answer and a suspicious request to logs of their own, so `error.log` keeps only the real faults:

Log (in `var/log/`)	What it holds	Level in the log viewer
`error.log`	real faults: a database error, a template that does not compile, a PHP error	error
`404.log`	answers 404 and 410, with the reason the kernel found	notice
`notice.log`	answers 401 and 403 (access answers)	notice
`security.log`	requests the request classifier names a probe, a traversal, an injection, an attack tool or a burst of not-found answers, with any status	warning

Each request is written to one of them, never to two: a probe that ends in a 404 is in `security.log` only. Nothing is blocked; the classifier only classifies (Request logs, in short).

> **Note**
> On a site where the web server (or a shield in front of it) refuses a request before

> Exponential starts, that request is not in these logs at all. Look in the web server's own logs for it.

Reading a 404.log line

```
[ Oct 17 2026 09:00:04 ] [2001:db8:x:x] 404 GET /content/view/full/9999 siteaccess=site reason=no
de:9999 node=9999 referrer=https://example.org/news ua=5d41402a "Mozilla/5.0 (Macintosh; Intel Ma
c OS X 14_5)"
```

Field	Meaning
`[2001:db8:x:x]`	the client address, masked
`404`	the status: 404 or 410
`GET /content/view/full/9999`	method and URI; every query value is replaced by ***
`siteaccess=`	the siteaccess that answered
`reason=`	why the page is missing: `module:<name>`, `view:<module>/<view>`, `disabled:<module>`, `node:<id>`, `object:<id>`, `rule:<name>`, `file:<object>/<attribute>`, or `kernel:<type>/<number>`
`node=`	the node id when known
`referrer=`	the page that linked here, or -
`ua=5d41402a "..."`	8 hex digits of the SHA-1 of the user agent and its first 60 characters

The `reason=` field is what makes the log useful to support: `node:9999` is a link to content that was removed, `module:wp-admin` is a robot looking for another product, `disabled:...` is a view the siteaccess switches off.

Privacy by default

- **Addresses are masked** (`[RequestLogSettings] MaskIP=enabled`): `203.0.x.x`, `2001:db8:x:x`. "The raw address is never written to a file." `disabled` writes it in full; the feature page says to do that "only for a site that has a reason and a legal basis for it".
- **Query values** are `***` in `404.log`; secrets and e-mail addresses in the path are masked.
- **The user agent** is a short hash plus its first 60 characters: "enough to tell one tool from another, not a fingerprint".
- The log viewer masks what it shows again, and the error digest is always masked.

> **Note**
>
> Masking reduces personal data in the logs; it does not decide how long you may keep them. That is part of your and your customer's data protection policy. This page is not legal advice; see Log and data retention and take your own legal counsel.

Settings

`site.ini [RequestLogSettings]` (override in `settings/override/site.ini.append.php`):

Setting	Shipped	Meaning
`NotFoundLog`	`enabled`	`disabled` sends not-found answers to `error.log`, as before
`NotFoundLogFile`	`404.log`	the file name, in the log directory
`MaskIP`	`enabled`	mask the client address
`MaxURILength`	`300`	the longest URI and referrer written; longer ones end in `...`

The log directory is `var/log`, or the site's own log directory with `[FileSettings] UseGlobalLogDir=disabled` (Storage layout). The classifier is configured in `settings/requestclassifier.ini`; see Security log.

Disk use and rotation

Both logs are kernel logs of the log viewer, with rotation settings of their own in `settings/logviewer.ini [RotationSettings]`: 5 archives each, compressed, a size limit of 10 MB for the cronjob part `logs`, and a prune that removes archives of `404.log` after 30 days and of `security.log` after 90 days. While a line is written, the writer also rotates the file at `site.ini [FileSettings] LogFileMaxSize` like every other log (Log rotation).

INI

```ini
# settings/override/logviewer.ini.append.php: keep security.log longer
[RotationSettings]
Keep[security]=10
MaxAgeDays[security]=365
```

The prune runs when the cronjob part `logs` runs (it is in no default part) or when `./console exp:logs --apply --prune` is called; see Log viewer and rotation.

Using the logs in support

Customer question	Where to look
"Visitors report broken links"	**Setup > Log files > Error digest**, `404.log` grouped by `reason=`; `node:#` entries are removed content, the `referrer=` names the page that still links to it
"Google reports many 404s"	the same digest; many `module:` reasons for paths of other products are robots, not broken links
"Is someone attacking us?"	`security.log` grouped by `<category>/<rule>` (Security log)
"A page showed an error with a reference"	`error.log`, found by the reference (Log viewer and rotation)

From a shell, the same files can be searched with the log viewer's command:

Shell

```
./console exp:logs --search='reason=node' --file=var/log/404.log --rotations
```

Checklist

- ☐ `NotFoundLog=enabled` and `MaskIP=enabled` (the shipped values) unless the customer decided otherwise in writing.
- ☐ The cronjob part `logs` or a scheduled `exp:logs --prune` keeps archives within their age limits.
- ☐ Support staff know the error digest and its grouping by `reason=`.
- ☐ Retention of request logs is part of the customer's data protection documentation.

Related pages

- Security log
- Log viewer and rotation
- Monitoring overview
- Request shield and request rules
- Request logs: 404.log and security.log (feature reference)
- 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.