

# Reference Manual

*Logs Manager Pro for Magento 2.x — Extmag\_LogsManager*

2026-05-24

- 1 Module layout
- 2 Admin routes
- 3 System configuration reference
  - 3.1 General ( `extmag_logs/general/...` )
  - 3.2 WebAPI Logs ( `extmag_logs/webapi_logs/...` )
  - 3.3 GraphQL Logs ( `extmag_logs/graphql_logs/...` )
  - 3.4 Email Logs ( `extmag_logs/email_logs/...` )
- 4 Cron jobs
- 5 Database schema
  - 5.1 `extmag_webapi_log`
  - 5.2 `extmag_email_log`
- 6 ACL resources
- 7 Log rotation algorithm
- 8 WebAPI capture pipeline
- 9 Email capture pipeline
- 10 Performance and scale notes
- 11 Known limitations

# 1 Module layout

`Extmag_LogsManager` is the marketplace shell module ( `extmag/logsmanagerpro` ). It declares a sequence dependency on `Extmag_LogsCore` ( `extmag/logs-core` ) where the real implementation lives — admin controllers, UI components, cron handlers, log writers, plugin interceptors, source models, ACL resources.

Module	Composer name	Role
<code>Extmag_LogsManager</code>	<code>extmag/logsmanagerpro</code>	Marketplace shell. Pulls in the core module.
<code>Extmag_LogsCore</code>	<code>extmag/logs-core</code>	Implementation: controllers, UI, cron, plugins.
<code>Extmag_Core</code>	<code>extmag/core</code>	Top-level <code>Extmag</code> admin menu + shared utilities.

# 2 Admin routes

Front name: `extmag_logs` . Routes are declared under `etc/adminhtml/routes.xml` .

Route	Controller	Purpose
<code>extmag_logs/logs/index</code>	<code>Extmag\LogsCore\Controller\Adminhtml</code>	Monolog records grid.
<code>extmag_logs/webapi/index</code>	<code>Extmag\LogsCore\Controller\Adminhtml</code>	WebAPI + GraphQL log grid.
<code>extmag_logs/email/index</code>	<code>Extmag\LogsCore\Controller\Adminhtml</code>	Email log grid.

# 3 System configuration reference

Stores → Configuration → Extmag → Logs Manager Pro ( `section_id = extmag_logs` ).

## 3.1 General ( `extmag_logs/general/...` )

Field	Default	Source model
<code>active</code>	1	<code>Magento\Config\Model\Config\Source\Yesno</code>
<code>file_max_size</code>	20	<code>text ( validate-digits )</code> , unit: MB
<code>amount_archives_copies</code>	5	<code>text ( validate-digits )</code>
<code>reports_compress</code>	0	<code>Magento\Config\Model\Config\Source\Yesno</code>

## 3.2 WebAPI Logs ( `extmag_logs/webapi_logs/...` )

Field	Default	Source model
<code>enable</code>	0	<code>Yesno</code>
<code>secret_mode</code>	mask	<code>Extmag\LogsCore\Model\Source\SecretMode</code>
<code>secret_words</code>	password,token,secret,authorization,api_key,credit_card	<code>text (comma-separated)</code>
<code>allowed_methods</code>	empty (= all)	<code>Extmag\LogsCore\Model\Source\RequestMethod (multi)</code>
<code>endpoint_mode</code>	all	<code>Extmag\LogsCore\Model\Source\EndpointMode</code>
<code>specific_endpoints</code>	empty	<code>Extmag\LogsCore\Model\Source\RestEndpoint (multi)</code>
<code>excluded_paths</code>	empty	<code>textarea</code> , one per line
<code>cleanup_hours</code>	720	<code>text ( validate-digits )</code> , 0 disables

`SecretMode` options: off, mask (default), remove. `EndpointMode` options: all (default), specific. `RequestMethod` options: GET, POST, PUT, DELETE, PATCH.

### 3.3 GraphQL Logs ( extmag\_logs/graphql\_logs/... )

Field	Default	Source model
enable	0	Yesno
only_mutations	0	Yesno
excluded_operations	empty	textarea, one operation name per line
cleanup_hours	720	text ( validate-digits )

cleanup\_hours shares the WebAPI cleanup cron — both groups age out of the same extmag\_webapi\_log table.

### 3.4 Email Logs ( extmag\_logs/email\_logs/... )

Field	Default	Source model
enable	0	Yesno
cleanup_hours	720	text ( validate-digits ), 0 disables

## 4 Cron jobs

All three jobs live in the Magento default cron group.

Job	Schedule	Handler	Gating
extmag_logs_archive	5 0-23/4 * * *	Extmag\LogsCore\Cron\ArchiveLogs::execute	extmag_logs/general/active
extmag_webapi_logs_cleanup	0 2 * * *	Extmag\LogsCore\Cron\CleanupWebapiLogs::execute	extmag_logs/webapi_logs/cleanup > 0
extmag_email_logs_cleanup	5 2 * * *	Extmag\LogsCore\Cron\CleanupEmailLogs::execute	extmag_logs/email_logs/cleanup > 0

The archive job runs every four hours starting at \*:05. Cleanup jobs fire once daily at 02:00 (WebAPI + GraphQL) and 02:05 (Email).

## 5 Database schema

### 5.1 extmag\_webapi\_log

WebAPI REST + GraphQL request capture.

Column	Type	Notes
log_id	int unsigned, AI	Primary key.
request_method	varchar(10)	GET / POST / PUT / DELETE / PATCH .
request_url	text	Full request URL.
request_headers	mediumtext	Serialized headers after secret masking.
request_body	mediumtext	Request body after secret masking.
requestor_ip	varchar(45)	Client IP, IPv4 or IPv6.
response_code	smallint unsigned	HTTP response status. Nullable on early exit.
response_headers	mediumtext	Response headers after secret masking.
response_body	mediumtext	Response body after secret masking.
duration	float	End-to-end seconds.
created_at	timestamp	Defaults to CURRENT_TIMESTAMP .

Indexes: created\_at , request\_method , response\_code .

## 5.2 extmag\_email\_log

Outgoing email capture.

Column	Type	Notes
log_id	int unsigned, AI	Primary key.
from_address	varchar(255)	Sender.
to_addresses	varchar(1024)	Comma-separated recipient list.
cc_addresses	varchar(1024)	Comma-separated CC list.
bcc_addresses	varchar(1024)	Comma-separated BCC list.
subject	varchar(512)	Email subject.
body_html	mediumtext	HTML body, as handed to transport.
body_text	mediumtext	Plain text body, as handed to transport.
status	varchar(20)	success / failed.
error_message	text	Transport exception message when status = failed.
created_at	timestamp	Defaults to CURRENT_TIMESTAMP.

Indexes: created\_at, status.

## 6 ACL resources

```
Magento_Backend::admin
├─ Extmag_LogsCore::logs                "Extmag Logs Manager Pro"
│  ├─ Extmag_LogsCore::logs_index      "View"
│  ├─ Extmag_LogsCore::logs_download  "Download"
│  ├─ Extmag_LogsCore::webapi_logs     "WebAPI Logs"
│  │  └─ Extmag_LogsCore::webapi_logs_clear "Clear All"
│  └─ Extmag_LogsCore::email_logs      "Email Logs"
│     └─ Extmag_LogsCore::email_logs_clear "Clear All"

Magento_Backend::stores
├─ Magento_Backend::stores_settings
│  └─ Magento_Config::config
│     └─ Extmag_Core::config
│        └─ Extmag_LogsCore::config     "Logs Manager Pro"
```

Each resource maps 1-to-1 with a menu item or with a destructive action on the matching grid. Role assignment is granular — read-only operators can be granted `logs_index` without the `*_clear` resources.

## 7 Log rotation algorithm

Extmag\LogCore\Cron\ArchiveLogs::execute (registered against `extmag_logs_archive`) does:

1. Read `general/active`. Bail if `0`.
2. Iterate `var/log/*` and, when `general/reports_compress = 1`, `var/report/*`.
3. For each entry:
  - Skip if it is already a `.zip`.
  - Skip if its file size is  $\leq$  `general/file_max_size` MB.
  - Create `<path>.<YYYY-MM-DD-HH-MM-SS>.zip` with `ZipArchive::OVERWRITE`.
  - `addFile` the source into the zip with its base name.
  - Truncate the source file to zero bytes.
4. After rotation, list `<base>*.zip` siblings ordered by mtime descending. Delete any past index `general/amount_archives_copies`.

`ZipArchive` errors bubble up to the cron framework and surface in the Magento `cron_schedule` table with `status = error`.

## 8 WebAPI capture pipeline

---

The WebAPI logger is attached via plugins on the WebAPI front controller and on the GraphQL request resolver. The capture path:

1. Plugin `before*` hook records request URL, method, headers, body and request start time.
2. Configuration is consulted before any database write:
  - `webapi_logs/enable` (or `graphql_logs/enable` for GraphQL) must be `1`.
  - `allowed_methods` multiselect must be empty or contain the method.
  - `endpoint_mode = specific` requires the resolved endpoint to appear in `specific_endpoints`.
  - `excluded_paths` lines are matched against the URL as substrings.
3. Plugin `after*` hook records response code, headers, body and computes `duration`.
4. Headers and body are passed through the secret-masking step:
  - `secret_mode = off` — no masking, raw values stored.
  - `secret_mode = mask` — values for keys whose name contains any `secret_words` token are replaced with a fixed mask string.
  - `secret_mode = remove` — matching keys are dropped from the serialized representation entirely.
5. A row is `INSERT` ed into `extmag_webapi_log`.

Capture failures are non-fatal — the original request / response is always served. Capture exceptions are logged into the `Magento exception.log` and the corresponding row is dropped.

## 9 Email capture pipeline

---

A plugin on the Magento mail transport intercepts `sendMessage` calls.

1. Plugin `before` resolves `From / To / CC / BCC / Subject / body / text body` from the message envelope.
2. Plugin `around` invokes the original `sendMessage`, catching any exception.
3. Plugin `row write`:
  - On success — `status = success`, `error_message = NULL`.
  - On exception — `status = failed`, `error_message` carries the exception message; the exception is re-thrown after the row is persisted so the caller still sees the failure.

The capture is gated by `email_logs/enable`. When the master switch is `0`, the plugin is still installed but exits immediately without touching the database.

## 10 Performance and scale notes

---

- The file rotation cron runs every four hours, not every minute — the scan cost (one `stat` per file under `var/log/`) is bounded.
- WebAPI and GraphQL capture writes one row per request. On high-traffic APIs (10k+ requests/hour) the `extmag_webapi_log` table grows fast; set `cleanup_hours` short enough that retention prune keeps up. The default `720` (30 days) is fine for moderate REST + GraphQL volume, not for a public storefront API.
- Cleanup runs nightly. Two large prunes in a row are possible after a retention reduction — expect a single longer cron tick the first time the new window applies.
- Email capture writes one row per outgoing message. Email volume is typically low enough that the default 30-day retention is fine.

## 11 Known limitations

---

- **Direct file writes** by third-party modules into `var/log/` are rotated, but if they write to a custom directory the cron does not see them. Move the path under `var/log/` or extend the cron.
- **Direct SQL writes** that bypass the Magento mail transport (transactional emails sent by raw `mail()` calls or by third-party modules that override transport) are not captured in the email log.
- **Multi-website scope** — the configuration is `default` only. There is no per-website override for retention, secret words, or enable flags.
- **GraphQL operation names** — when an operation is anonymous the filter `excluded_operations` cannot match it. Anonymous operations are always logged when `graphql_logs/enable = 1`.
- **Cron downtime** — if the host is down for longer than four hours the archival cron runs once when cron resumes and processes the current state. No catch-up batching.

- **Body size** — `mediumtext` columns cap individual bodies at 16 MB. Larger payloads are truncated by MySQL with a warning.