Logs

Logs

> *Dernière mise à jour : 2026-05-10*

Logger central

Tous les logs serveur passent par `src/lib/logger.ts`. Conventions :

- **Aucun** `console.*` dans `src/` (sauf `src/config.ts` qui boot avant le logger)
- Préfixer le scope : `logger.info('[meta-sync] ...')`, `logger.error('[ssh] ...')`
- Niveaux : `debug` / `info` / `warn` / `error` (filtrable via `LOG_LEVEL`)
- Format prod : JSON ligne (parseable via `jq`)
- Format dev : pretty print (lisible sans outil)

Locations

- **Stdout / stderr Docker** : `docker logs boardgame-referee`
- **Fichier persistant** : `/app/data/logs/server.log` (volume Unraid `data/`)

Rotation

Géré par `entrypoint.sh` au boot :

- Si `server.log > 50 Mo` → archive en `server-YYYYMMDD-HHMMSS.log`
- Purge automatique des archives `> 30 jours`

Pas de rotation runtime (only at boot). Si tu fais beaucoup de DEBUG en prod, pense à restart le container ou logrotate manuellement.

Filtrer

```
# Live tail
docker exec boardgame-referee tail -f /app/data/logs/server.log

# Live tail avec scope filter (jq pour le JSON)
docker exec boardgame-referee tail -f /app/data/logs/server.log | jq 'select(.scope == "rag")'

# Grep classique
docker exec boardgame-referee grep "claude-quota" /app/data/logs/server.log | tail -50

# Toutes les erreurs des 24 dernières heures
docker exec boardgame-referee grep -E '"level":"error"' /app/data/logs/server.log | tail -100
```

Niveaux par défaut

Environnement	`LOG_LEVEL`
Dev	`debug`
Prod	`info`

Pour passer en debug en prod (temporaire) :

```
# Modifier .env Unraid : LOG_LEVEL=debug
# Restart container :
docker compose -f /mnt/user/appdata/boardgame-referee/docker-compose.yml restart app
```

⚠ N'oublie pas de remettre en `info` après — debug est verbeux et remplit vite les 50 Mo.

Logs notables à surveiller

Pattern	Sens

`[claude-quota] ClaudeQuotaError`	Pause auto enclenchée, vérifier `resetAt`
`[ssh] command rejected`	Wrapper oracle a bloqué une commande (debug `/tmp/oracle-debug.log` si nécessaire)
`[ingest] stage error`	Stage d'ingestion en échec
`[meta-sync] purge`	Purge hebdo des chunks méta périmés
`[forum-sync]`	Cron forums BGG (3h du matin par défaut)
`[hyde] timeout`	HyDE Haiku a dépassé 25s, fallback question brute
`[contextual-cache] flushed`	Cache JSON sauvé sur disque (avant quota error ou checkpoint)
`[rag] retrieve`	Diagnostics retrieval (timings, scores) en debug

Pas de log aggregation externe

Pas de Loki / Grafana / ELK / Sentry actuellement. Tout reste dans le fichier Unraid + Docker logs.

Si un jour tu veux ajouter Sentry ou un log shipper :

- Ajouter une lib (`@sentry/node`)
- Wrapper le logger pour aussi pousser vers Sentry sur `level === 'error'`
- Configurer `SENTRY_DSN` env var
- Le code reste centralisé dans `src/lib/logger.ts` — les autres fichiers n'ont pas à savoir qu'on log ailleurs

Debug d'une question

Pour relire les diagnostics d'une question donnée (lecture seule) :

```
docker exec boardgame-referee npm run debug:question -- --id <questionId>
```

Affiche : question, réponse, chunks retrouvés, HyDE, timings — équivalent CLI du panneau `/admin/feedback/<id>`.

Revision #1
Created 2026-05-10 15:20:11 UTC by thymon
Updated 2026-05-10 15:20:11 UTC by thymon