

Crash console — open ingest API (design)

From OPEN_API.md

Field	Value
Author	Anton Afanasyeu
Revision	R1
Creation date	2026-05-23
Last modification date	2026-05-23
Co-authored	
Severity	medium
State	in progress
Document type	technical

Table of contents

[Problems](#)
[Payload extension \(ingest\)](#)
[Upload API \(POST .../api/upload.php\)](#)
[Console / query API \(read\)](#)
[Classifying synthetic traffic \(heuristics\)](#)
[Security](#)
[Android](#)
[Triage tags \(implemented v1\)](#)
[Implementation order](#)

Source: OPEN_API.md — section links work in PDF viewers that support internal anchors.

Triage **real device crashes** vs **synthetic / bulk test traffic** when exposing upload or query APIs beyond the Android app. Builds on RBAC (examples/crash_reporter/backend/README.md).

Problems

- Load tests and CI may POST many similar JSON blobs → drown signal in the reports list.
- Third-party integrations need stable auth and filters without console login cookies.
- Similarity grouping already uses payload fingerprints; need **provenance** to exclude known harness traffic.

Payload extension (ingest)

Add optional top-level fields on upload (backward compatible):

Field	Type	Meaning
ingest.source	string	device (default), open_api, ci, manual
ingest.client_id	string	Stable id per API key or CI job
ingest.run_id	string	Correlates a burst of reports from one test run

Android app omits these (defaults to device). Harness sets open_api + run_id.

Store in DB:

- Column reports.ingest_source (indexed) **or** JSON path in payload_json with generated column later.
- Prefer column for list filters and MariaDB permissions story.

Upload API (POST .../api/upload.php)

Clients may send Content-Encoding: gzip with a gzip-compressed JSON body (same schema). The BE nginx upload location should use gunzip on + gunzip_types application/json; PHP accepts plain or gzip.

Phases:

1. **Phase A (now)**: Accept extra JSON fields; persist inside payload_json; console search includes ingest.source in blob.
2. **Phase B**: API key header X-Crash-API-Key → maps to company_id (RBAC); rate limit per key.
3. **Phase C**: Reject or quarantine when ingest.source=open_api and fingerprint matches org blocklist.

Console / query API (read)

Thin JSON endpoints (session or API key):

Endpoint	Purpose
GET /api/reports?ingest_source=device	Default triage view
GET /api/reports?ingest_source_ne=open_api	Hide harness
GET /api/reports?group=fingerprint&min_count=N	Bulk duplicate detector
GET /api/reports/{id}	Detail (existing shape)

UI: filter chips on reports toolbar — **Devices only** / **Include API traffic** / tag synthetic.

Classifying synthetic traffic (heuristics)

Mark report or tag synthetic when any:

- ingest.source in (open_api, ci) **and** user has not promoted it.
- Same ingest.run_id + fingerprint count > threshold in 10 minutes.
- Identical report_id prefix pattern from known test fixture.

Manual: console **Clear synthetic tag** / **Promote to real** on detail page.

Security

- API keys scoped to company_id; no cross-tenant list.
- Separate write-only keys for CI upload vs read keys for dashboards.
- Log upload failures to existing crash event log; no stack traces in response body.

Android

No change required for v1. Optional later: dev setting to set ingest.source>manual for dogfood builds.

Triage tags (implemented v1)

Workflow labels (multi-tag per report, any combination):

id	Meaning
----	---------

in-progress	Someone is working on it
fixed	Fix landed
rejected	Won't process
done	Closed / EOL for this issue
not-reproducible	Cannot reproduce
not-a-bug	Expected behaviour
duplicate	Duplicate of another report

API

Endpoint	Purpose
GET /api/tag_catalog.php	Workflow presets + editor suggestions
GET /api/report_tags.php?id=N	Tags on one report
POST /api/report_tags.php	Body {"id":N, "tags":[...]} — replace custom tags
GET /api/reports.php?tag=fixed&tag=duplicate&tag_mode=and	List filter (AND default, OR optional)

Smoke test: `examples/crash_reporter/backend/scripts/test_tags_api.sh`

Implementation order

1. ~~Workflow tags + list filter~~ (done)
 2. Document fields; parse and store in payload (Phase A ingest).
 3. List filter `ingest_source` in `ReportRepository` + UI chip.
 4. API keys table + `x-Crash-API-Key` on upload.
 5. Grouped bulk endpoint for duplicate review.
- See also `CRASH_REPORTER.md` and `examples/crash_reporter/ROADMAP.md`.