Proofline

EXAMPLE DATA · SAMPLE REPORT

Load, API & Security Test Report

Example Pay API v2 — fictional fintech system

Client	Example Pay Ltd (fictional)
Scope	48 REST endpoints, staging environment
Test types	Load, stress, spike · functional & contract · OWASP API Top 10
Version	1.0 · sample

Every system name, finding and figure in this document is fictional and labelled EXAMPLE DATA. It shows the structure and depth of a Proofline report, not a client result.

1. Executive summary

~560 VUs	2	9	6 / 10
max. sustained load within SLO	high-severity findings	findings in total	OWASP API categories affected
EXAMPLE DATA	EXAMPLE DATA	EXAMPLE DATA	EXAMPLE DATA

Verdict

The API is **not ready** for the planned launch traffic of 800 concurrent users. Two issues must be fixed first: a broken object-level authorisation flaw that exposes other tenants' account data (F-01), and connection pool exhaustion that pushes p95 latency past the 400 ms SLO at about 600 virtual users (F-02).

Top three actions

1. Enforce tenant ownership checks in the accounts service (F-01). Effort: S.
2. Raise DB pool size and add pgbouncer; re-test at 800 VUs (F-02). Effort: M.
3. Add per-key rate limiting on webhooks with 429 + Retry-After (F-03). Effort: S.

Release recommendation

Go-live after F-01 and F-02 are fixed and verified by re-test. Medium findings can follow within the first two sprints.

2. Scope, environment & method

ITEM	DETAIL
System under test	Example Pay API v2 (REST, OpenAPI 3.1)
Environment	Staging, production-equivalent sizing (3 app nodes, managed PostgreSQL)
Test data	Synthetic: 2 tenants, 5,000 accounts, 200,000 transactions
Tools	`k6 · Grafana · Postman/Newman · Schemathesis · Burp Suite`
Period	10 working days (example)
Out of scope	Mobile apps, third-party card processor, infrastructure penetration test

Load model

SCENARIO	PROFILE	PASS CRITERIA
Load	ramp to 200 VUs, hold 10 min	p95 < 400 ms · errors < 1%
Stress	+100 VUs every 2 min to failure	
Spike	200 → 800 VUs in 30 s	
Soak	300 VUs for 4 h	

Severity scale

`High` data exposure or SLO breach at expected load · `Medium` degradation or exploitable with effort · `Low` hygiene, low impact

3. Performance results

Fig. 1 — Latency vs. virtual users, POST /v1/payments (stress run) EXAMPLE DATA

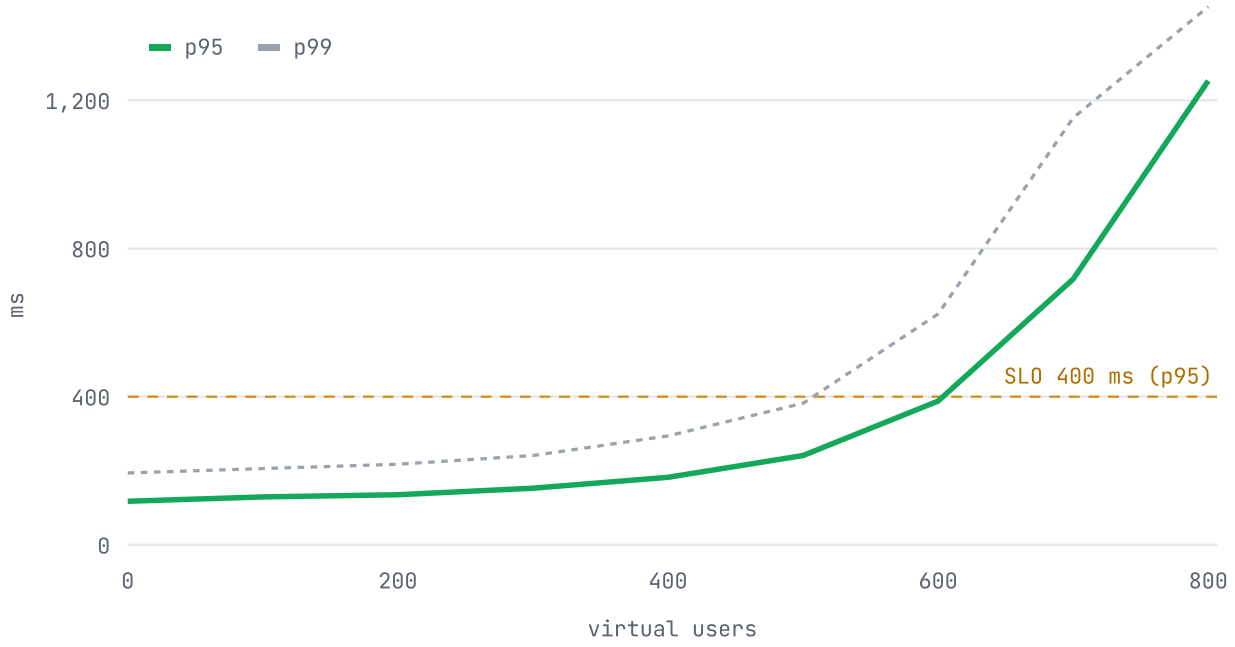
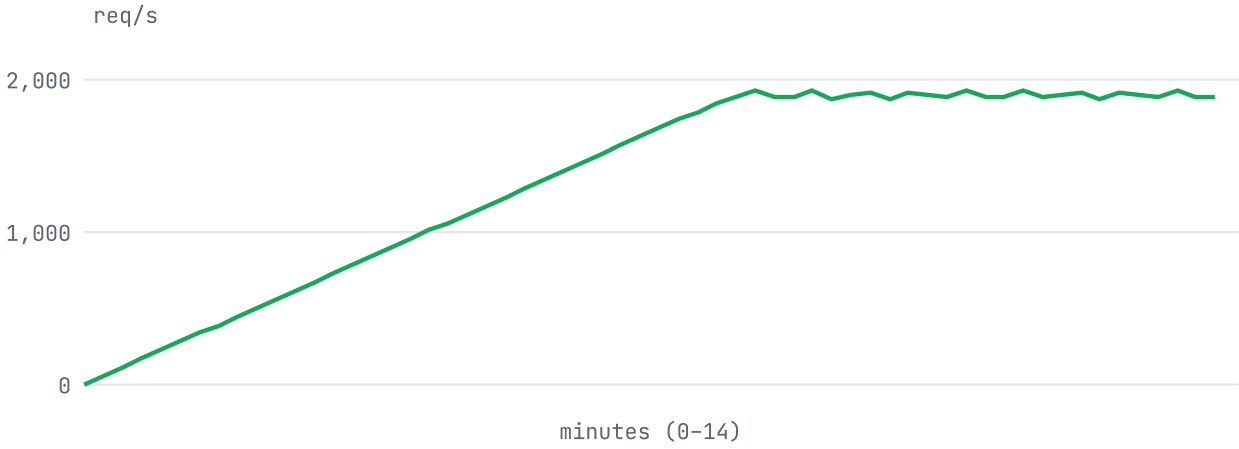


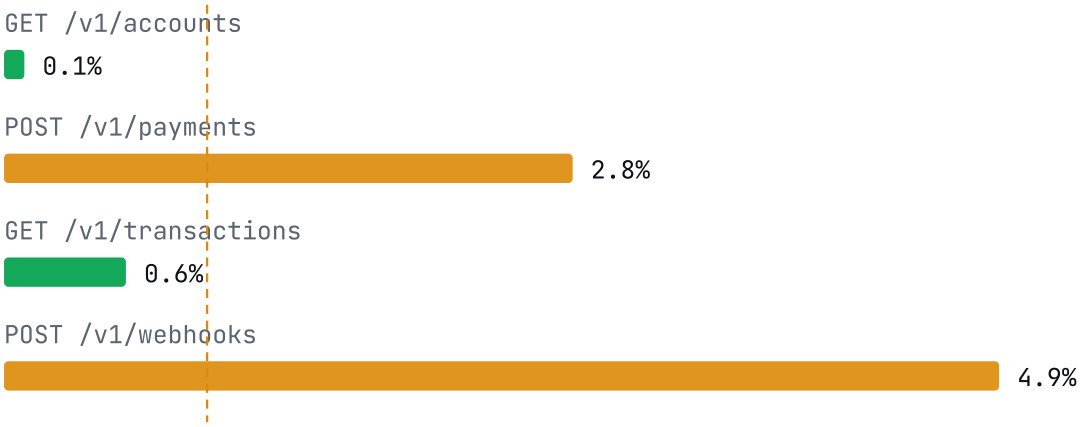
Fig. 2 — Throughput during stress run EXAMPLE DATA



p95 latency stays flat up to ~400 VUs, then rises sharply. The SLO is breached at ~600 VUs (p95 390 → 720 ms). Throughput plateaus at ~1,900 req/s, consistent with a saturated resource rather than CPU limits (app CPU peaked at 58%).

3. Performance results (cont.)

Fig. 3 — Error rate per endpoint at 600 VUs (threshold 1%) EXAMPLE DATA



SCENARIO	P95	P99	ERRORS	RESULT
Load (200 VUs)	135 ms	210 ms	0.1%	Pass
Stress (to failure)	1,250 ms	2,030 ms	3.4%	Fail at ~600 VUs
Spike (200→800)	1,480 ms	2,600 ms	6.2%	Fail
Soak (4 h, 300 VUs)	160 ms	260 ms	0.2%	Pass, no memory growth

All values are EXAMPLE DATA.

4. Findings overview

ID	SEVERITY	CATEGORY	FINDING
F-01	High	API1:2023 BOLA	Cross-tenant read on GET /v1/accounts/{id}
F-02	High	Load	DB connection pool exhausted at ~600 VUs
F-03	Medium	API4:2023	No rate limit on POST /v1/webhooks
F-04	Medium	Contract	`amount` returned as string, spec says number
F-05	Medium	API2:2023	Refresh tokens not revoked on password change
F-06	Medium	Load	Payment idempotency key ignored under retry storm
F-07	Low	API8:2023	Verbose stack traces in 500 responses
F-08	Low	QA	Date picker accepts 31 February
F-09	Low	Contract	Undocumented 409 status on POST /v1/payments

OWASP API Security Top 10 (2023) coverage

API1 BOLA	F-01	API6 Sensitive business flows	F-06
API2 Broken Auth	F-05	API7 SSRF	tested, none
API3 Property-level authZ	tested, none	API8 Misconfiguration	F-07
API4 Resource consumption	F-03	API9 Inventory management	F-09
API5 Function-level authZ	tested, none	API10 Unsafe API consumption	tested, none

5. Finding detail (excerpt)

F-01 · High Cross-tenant read on GET /v1/accounts/{id}

Impact: Any authenticated user can read account balances and IBANs of other tenants by changing the account ID.

```
# Authenticated as tenant A
curl -s -H "Authorization: Bearer $TOKEN_TENANT_A" \
  https://api.example.test/v1/accounts/acc_B_7731 | jq '.tenant_id'
# expected: 404 · actual: "tenant_b"
```

Fix: Scope every account query by the tenant ID from the token; return 404 for foreign IDs. Add a negative contract test to CI.

F-02 · High DB connection pool exhausted at ~600 VUs

Evidence: At ~600 VUs, pool wait time rises to 480 ms (p95) while DB CPU stays at 41%. 20 of 20 connections in use for the whole plateau.

Fix: Raise pool size to 50 per node, add pgbouncer in transaction mode, re-test stress and spike at 800 VUs.

F-03 · Medium No rate limit on POST /v1/webhooks

Evidence: Burst of 5,000 requests in 10 s accepted; 4.9% returned 5xx, no 429 issued.

Fix: Per-key token bucket; respond 429 with Retry-After.

6. Remediation plan & re-running the tests

PRIORITY	FINDINGS	EFFORT	WHEN
P1	F-01, F-02	S + M	Before launch
P2	F-03, F-05, F-06	S, S, M	Sprint 1
P3	F-04, F-07, F-08, F-09	S	Sprint 2

Deliverables in your repository

```
tests/
  load/      checkout-load.js  stress.js  spike.js  soak.js
  api/       newman-collection.json  schemathesis.yml
  security/  bola-checks.sh  ratelimit-burst.js
.github/workflows/perf-smoke.yml
```

Re-run

```
k6 run -e BASE=https://staging.example.test tests/load/stress.js
newman run tests/api/newman-collection.json
schemathesis run openapi.yaml --checks all
```

A re-test of P1 findings is included after fixes are deployed. Debrief: 60 minutes with your engineering team.