@demotoprod · SHIP · also LEVERS

Rate limit checklist

A 429 response means the user has sent too many requests in a given amount of time, and a client that retries at once sends more. One engineering post shows 3 retries at each of 5 layers raising database load 243x. Five checks, each sized for an afternoon.

1 Wait for Retry-After, then add jitter

RFC 6585 says a 429 response MAY include a Retry-After header saying how long to wait. The header is optional, so a client needs its own delay too. In one published simulation, adding jitter to exponential backoff cut the call count by more than half with 100 contending clients.

- Wait at least the Retry-After seconds plus a small random delay, so clients do not return together.
- Without the header, wait a random time between 0 and a capped exponential delay.

```
import random

def next_delay(attempt, retry_after=None, base=1.0, cap=60.0):
    if retry_after is not None:        # seconds, from the Retry-After header
        return retry_after + random.uniform(0, 1)
    return random.uniform(0, min(cap, base * 2 ** attempt))
```

2 Pace your own calls with a token bucket

One provider documents 60 requests per minute as a rate that might be enforced as 1 per second, so short bursts can trigger errors, and describes capacity as a token bucket refilled continuously. A second provider's guide says unsuccessful requests count toward the per-minute limit.

- Put a token bucket in front of every call, refilled below your published limit, one bucket per limit (requests and tokens per minute).
- After a 429, lower the rate and climb back in steps.

3 Trip a circuit breaker and route to a fallback provider

A circuit breaker wraps a protected call and trips once failures reach a threshold; further calls fail without the protected call being made, sparing a struggling server, and after an interval a trial call goes through (half open).

- Give each provider its own breaker with a written threshold and open interval, held open at least Retry-After.
- While open, route to the fallback provider or a cached answer; close only after a successful trial call.

4 Cap retries per request, per client and per layer

With 3 retries at each layer of a 5-deep stack, load on the database rises 243x. A published overload design budgets 3 attempts per request and retries only while retries are below 10% of requests; with both, growth falls to 1.1x in the general case.

- Retry at one layer only and pass the error up from the others.
- Cap attempts per request and the retry share of requests; alert on the ratio.

5 Retry only calls that are safe to repeat

A retry can run an operation more than once. One post says APIs with side effects are not safe to retry unless they provide idempotency; read-only APIs are typically idempotent, resource creation may not be, and some APIs offer token-based mechanisms that make them safe.

- List every retried call and mark it read-only, idempotent or neither.
- For neither, send an idempotency token if the API accepts one, otherwise do not retry.

Nitish Gautam · indica-tech.com · hello@indica-tech.com

INDICA TECH · @DEMOTOPROD

Nitish Gautam

Founder-led AI engineering. Demo to production, under NDA, in regulated industries, on fixed scope.

Paste into your project

A RETRY POLICY TO ADAPT PER PROVIDER

```
# One block per provider; names in capitals are yours to set
retry:
  on_status: [429]
  wait: retry_after_plus_jitter  # else random 0 to capped backoff
  max_attempts: 3                # per request, one published design
  max_retry_ratio: 0.10          # 10% of requests, same design
  layer: top_only                # lower layers pass the error up
  only_if: [read_only, idempotent, idempotency_token]
pacing:
  token_bucket_rate: RATE_BELOW_PUBLISHED_LIMIT
breaker:
  failure_threshold: FAILURE_THRESHOLD
  open_seconds: AT_LEAST_RETRY_AFTER
  fallback: [second_provider, cached_answer]
```

Where this list comes from

Ten-plus years taking products from zero to one across seven sectors, including defence, fintech, medical and telecom, with AI shipped into regulated and air-gapped environments. The @demotoprod channel takes one production AI failure apart each day. These checklists are the same work written down: every item is a dated failure or a documented default, with the control that stops it and the primary source.

If this list found a gap

1. ### Your demo works and production is stuck.

   Production Readiness Audit. Fixed price, two weeks. A five-dimension scorecard, cost and failure analysis, and a 90-day execution roadmap.

2. ### You need senior AI judgement without a full-time hire.

   Fractional AI Lead. Monthly: 90-day minimum, then 30 days' notice either side. Weekly 1:1 with the founder or CTO, architecture and code review, technical hiring support, investor-ready due diligence.

3. ### An AI feature has to work, by a date.

   AI Build Sprint. Fixed price, four to eight weeks. Scope signed off up front, weekly demos, CI/CD and observability included, 30-day production warranty.

4. ### A regulator, an auditor or a customer's security team is asking.

   AI Governance & Compliance. Fixed price, four weeks. ISO 27001 / SOC 2-aligned controls, a model risk framework, human-in-the-loop and audit logging, regulator-ready documentation.

BOOK A CALL
indica-tech.com/#contact

EMAIL
hello@indica-tech.com

WEBSITE
indica-tech.com

ALL CHECKLISTS
indica-tech.com/checklists/

YOUTUBE
youtube.com/@demotoprod

LINKEDIN
linkedin.com/in/startup-founderx

INSTAGRAM
instagram.com/demotoprod

TIKTOK
tiktok.com/@demotoprod

Free to share. The live list, with every source link: indica-tech.com/checklists/#rate-limit