The Idempotency-Key HTTP Header Field



Dmitrii Tishkin · HTTPAPI WG · IETF 127 · November 2026

What's new in -08?

- Draft revived (-07 expired in April 2026)
- Updated references to obsoleted RFCs: RFC 7231 to RFC 9110, RFC 7807 to RFC 9457, RFC 4122 to RFC 9562 and RFC 8941 to RFC 9651
- Retry case (#48): adopted the proposed text, still open (see later slide)
- Editorial fixes; removed unused ABNF and IMF-fixdate reference
- New author: Dmitrii Tishkin

Merged, will be in -09

- #34 Key uniqueness → PR #60: key is opaque; a resource may define uniqueness (e.g. per tenant); exact match by default
- #36 Fingerprint → PR #63: stays optional; explains its purpose and how matching/different fingerprints are handled
- #37 Resource requirements → PR #61: documentation MUST → SHOULD

Response to a retry (#48, #46)

- -08 applied the text proposed in #48 (PR #53): “a response consistent with the request being completed according to the user’s intent”
- Follows RFC 9110 (same effect, response may differ) and still allows replaying the stored response
- Open: do errors count as “completed”? Practice differs; #48 suggests a retry after a 4xx is processed as new
- #46 replay indicator: not proposed. Its main use case is a generic intermediary, which is out of scope (#46)

Q1: Does the WG accept the -08 text?

Q2: Should the resource’s own errors be replayed on retry?

Error responses: the problem (#42, #51)

- The draft uses 400 (key missing), 422 (key reused with a different payload), 409 (original request still in progress)
- APIs already use 409/422 for their own errors, so a client can't tell "retry later" from "don't retry"
- All three examples share one type URI, so even Problem Details can't tell them apart

Proposal: register problem types

- Three types under https://iana.org/assignments/http-problem-types#, as in digest-fields-problem-types:
  1. idempotency-key-missing (400)
  2. idempotency-key-reused (422): don’t retry as is
  3. idempotency-request-in-progress (409): retry later, MAY send Retry-After
- Status codes unchanged; clients identify the error by type
- API documentation moves to Link: rel="describedby"; a Link-only response (no body) is no longer an alternative

Q: Does the WG support this direction? Names are open

Key expiry (#35, #49)

- Today: resource MAY use “time based” keys and SHOULD publish an expiration policy. Nothing else
- Missing: what happens after expiry, and guidance on retention (practice: 1 hour to 7 days)

Proposal: define what expiry means

- Scope: the mechanism protects retries within the retention period; after it, a retry is processed as new
- Resources SHOULD retain keys for as long as clients are expected to retry, and document the period; Clients SHOULD NOT retry with the same key after it
- Non-normative guidance: what to store per key, when the period starts, example periods (e.g. 1 h, 24 h, 72 h); no normative min/max
- Out of scope: retention-period header (lost along with the response), first-sent timestamp (OASIS)

Q: Does the WG agree with this direction?

Relationship to OASIS Repeatable Requests (#31)

- #31 asks: what does Idempotency-Key cover that RR doesn't? A crisp answer is needed; no comparison section in the draft
- Clients can send Idempotency-Key opportunistically, without knowing if the server supports it. RR needs coordination (servers that can't comply MUST return 501)
- Fewer constraints: one header. RR prescribes response codes, expects transactional consistency, and requires a first-sent timestamp from clients
- Already deployed under this name (Stripe, Adyen, WorldPay, Yandex, ...)

Q: Is this answer crisp enough to close #31?

Next steps

- PRs #60, #61, #63 merged → in -09; #34, #36 closed; #37 stays open (rest in #42, #48)
- Turn today's outcomes into PRs:
  1. problem types for errors (#42, #51)
  2. expiry text and guidance (#35, #49)
  3. errors on retry (#48, Q2)
- Close #46 and #31 if the WG agrees; record the answer to #31 in the issue
- Confirm all decisions on the mailing list
- Publish -09 after IETF 127 → work towards WG Last Call
- Feedback: httpapi@ietf.org · github.com/ietf-wg-httpapi/idempotency

Thank you!