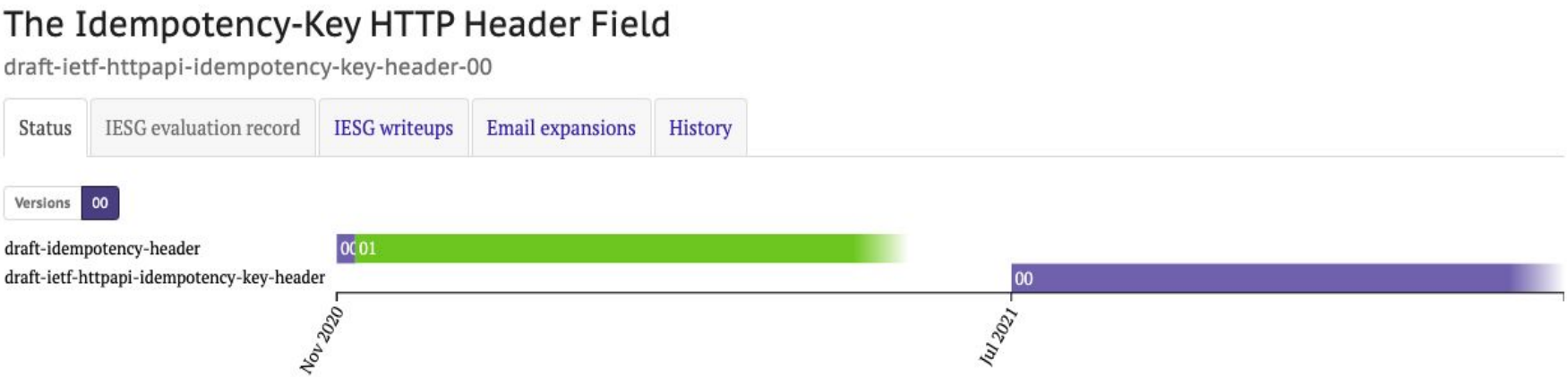
Idempotency Key Header

Jayadeba Jena jjena@paypal.com
Sanjay Dalal sanjay.dalal@cal.berkeley.edu

IETF 111
July 26, 2021

History



- First ID Nov 2020
- Accepted by IETF HTTP API WG: Jul 2021

Motivation

I-D:
https://github.com/ietf-wg-httpapi/idempotency/blob/main/draft-ietf-httpapi-idempotency-key-header.md

Authors: Jayadeba Jena, Sanjay Dalal, Erik Wilde

- POST (and PATCH) is not idempotent but widely used. Idempotency for POST is a common use case across many HTTP APIs, esp. in cases where the cost of duplicate processing is quite high such as money transfer transactions.
- Some API developers have named an HTTP header as Request-Id, some define their own header (e.g. PayPal-Request-Id), some use x-idempotency-key (OpenBanking/PSD2) and others embed idempotency-key in request message body
- A simple HTTP standard header would make interoperability possible and increase developer mindshare. Tools could be built to process it automatically.

Implementers

- Implementers: Stripe, Adyen, Dwolla, Interledger, WorldPay, Yandex, http4s.org, Finastra, Datatrans
- Different header: PayPal, Django, Twilio, RazorPay, OpenBanking, BBVA
- Implementing the concept: Google Standard Payments, Square

Status

Issues: https://github.com/ietf-wg-httpapi/idempotency/issues

- #2 Clarification for status code for various scenarios
- #3 Feedback from Google Standard Payments
- #4 Conditional requests RFC 7232
- #5 How does this header compare with OASIS Repeatable Requests Header?

Issue #2

- Whether to return 200, 204 or 422
- Suggestions on softening the language (MUST, MAY, SHOULD, …)
- Is idempotency fingerprint a must for implementation?

Issue #3

- Google Standard Payments prefers protocol agnostic solution
- Separating the idempotency-key from the rest of the payload doesn't make sense

Issue #4

- Could use conditional requests as described in RFC 7232 for idempotency
- Suggestion is to refer this alternative solution in appendix

Issue #5

- One of the chairs of OASIS Open Data Protocol reached out with a proposal of alignment between Idempotency-Key header and OASIS Repeatable Requests Header
- “The two headers Idempotency-Key and Repeatability-Request-ID seem to have identical semantics. I see potential in aligning these two proposals.”

Thanks!

Idempotency Key Header (Header: “Idempotency-Key”)

- MUST be unique for every request from a particular client.
- Has an expiry time (purged or deleted by the server after the key is expired. Expiry time is defined by the server and published in the documentation).
- Key can’t be reused with another request within the expiry time
- UUID v4 or similar is recommended as the idempotency key

Idempotency Fingerprint

An idempotency fingerprint MAY be used in conjunction with an idempotency key to determine the uniqueness of a request. The server may use one of the following algorithms to generate a fingerprint.

- Checksum of the entire request payload.
- Checksum of selected element(s) in the request payload.
- Field value match for each field in the request payload.
- Field value match for selected element(s) in the request payload.
- Request digest/signature.