What is idempotency, and why does it matter for APIs?

1 November 2026 · 4 min read · blogs.iictp.org/what-is-idempotency

An operation is idempotent if doing it several times has the same effect as doing it once. This matters because networks fail: a client that times out cannot tell whether its request arrived, so it retries, and the system must handle the repeat safely.

TL;DR The short version

- An operation is idempotent if doing it several times has the same effect as doing it once, which matters because clients retry after timeouts.
- GET, PUT and DELETE are idempotent by definition; a retried POST can create two orders or two charges.
- Make POST safe with a client-generated idempotency key the server remembers for hours to days, and make message consumers record what they processed.

Read online, listen to it, or change the text size: https://blogs.iictp.org/what-is-idempotency

Which HTTP methods are idempotent?

GET, HEAD, PUT, DELETE and OPTIONS are defined as idempotent, while POST and PATCH are not by default. Idempotent describes the state on the server, not the response code: a repeated DELETE may return 404 and still be idempotent.

Why is a repeated POST dangerous?

A POST that creates an order or charges a card can create two orders or two charges when a client retries after a timeout, because the server sees two separate requests.

How do you make a POST safe to retry?

The client generates a unique key for each logical operation and sends it with the request, usually in a header. The server stores the key with the result of the first request, and on a repeat it returns the stored result without doing the work again.

How long should the server remember keys?

Long enough to cover any realistic retry, which is usually hours to days, and then it can discard them. The server should also reject a reused key that arrives with a different request body, since that signals a client bug.

Where else does idempotency appear?

Message consumers also receive some messages more than once, so they should record what they have processed. Deployment tools aim for idempotence too: applying the same configuration twice should leave the system unchanged.

> Related credential: Associate Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. IETF. RFC 9110, HTTP Semantics, section 9.2 (Common Method Properties). rfc-editor.org/rfc/rfc9110.

Published by IICTP, an independent technical credential body, on 1 November 2026. The current version of this post is at https://blogs.iictp.org/what-is-idempotency. More guides: blogs.iictp.org.