When to retry

idempotent operation

no matter how many times this operation is performed, the result is always the same

idempotent

x = 5

not idempotent

x++

idempotent API

making multiple identical requests to that API has the same effect as making a single request

read API is typically idempotent

| service A | read data request → | service B |

write API is typically not idempotent

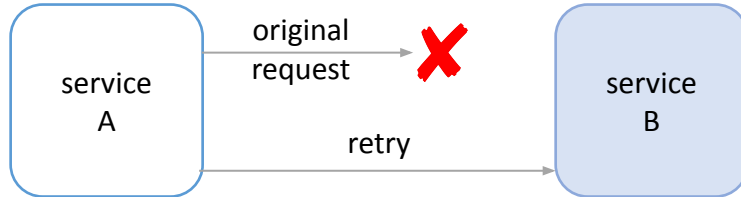| service A | create a new record request → | service B |

When to retry

✔️ idempotent APIs are safe to retry

Can we retry a failed request to a non-idempotent API?

yes, if

original request never hit the API

service A → original request ❌ → service B

service A → retry → service B

API supports idempotency keys (client tokens)

service A → original request (idempotency key) → service B

service A → retry (same idempotency key) → service B

> I will store the idempotency key for several minutes (hours) only.
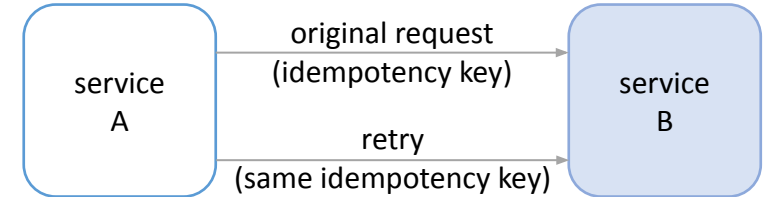
software engineer

> I need to document which APIs are idempotent and share this information with all clients of my service.

When to retry

HTTP Status Code	Description	Retry?	Comments
N/A	Network connection is not established.	Yes	It doesn't matter if the API is idempotent or not. If no connection was established, the request was never sent.
400	Bad request.	No	There is a problem with the request. For example, a required parameter is not supplied for the specified action. Retry will not help.
403	Access denied.	No	One of many possible reasons - the provided security credentials are not valid.
404	Not found exception.	No	There is no requested resource present on the server side. E.g. the specified S3 bucket does not exist.
429	Too many requests.	Yes (but after a delay)	The request was not processed. It was immediately rejected by the server due to the high request rate.
500	Internal server error.	Yes (if idempotent)	The service encountered an unexpected condition that prevented it from fulfilling the request.
503	Service unavailable.	Yes	The requested service is not available.
504	Request timeout.	Yes (if idempotent)	The request hit the server, but could not be completed within the timeout period.