



**#ASLI ENGINEERING**

# Implementing Idempotence in Payments Service



**BY**

**ARPIT BHAYANI**

# Idempotence in Payments Service

execute same operation multiple times,  
result is same as if operation was applied just ONCE



No matter how many times  
user likes one post, the like count  
should just increase by 1

♥ 729 → ♥ 730

The situation becomes even more serious when \$ is involved



Say A wants to transfer \$ 20,000 to B  
and if due to any reason the API call  
is retried.

we would not want

A to transfer twice the amount to B.

We need idempotence

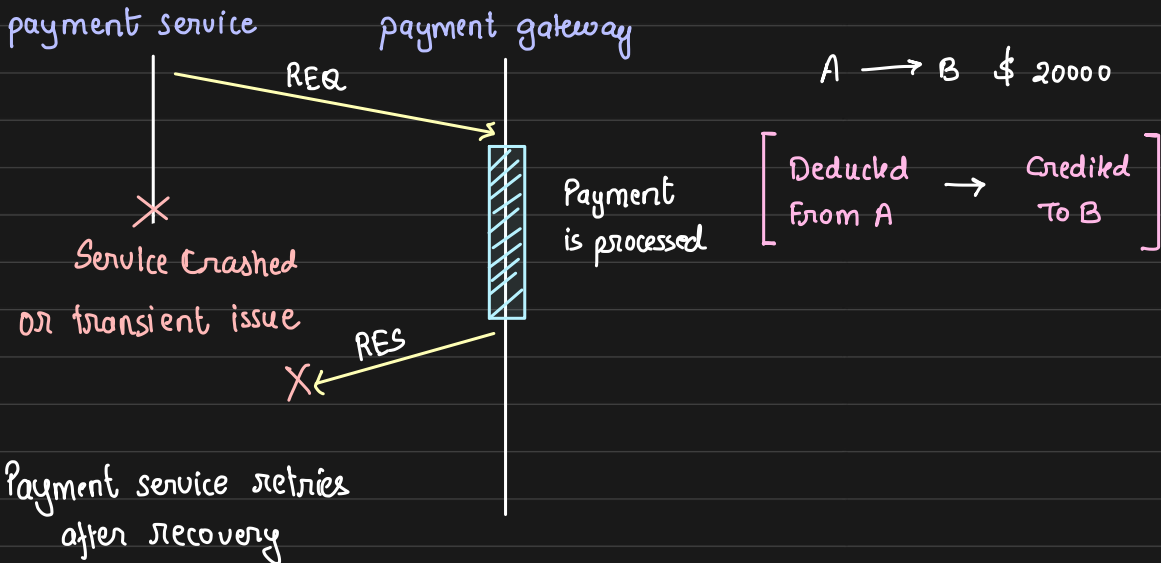
Other places where this is relevant

- user to not tweet the same tweet again
- not placing the same order twice on Amazon
- not sending the same message again

Why would the transaction repeat?

user or service

You **retry** when something goes wrong



payment service

payment gateway

REQ

X

Service Crashed

Payment  
is processed

A → B \$ 20000

[ Deducted From A → Credited To B ]

RES

X

But your service doesn't know  
hence it retries !!

SAME REQ

Payment  
is processed  
again!

Retry only when  
you are sure

RES

let's see how can we implement idempotence!

## Implementing Idempotence

Approach 1: You won't need idempotence if we Do not retry

Depending on your product/usecase  
this might be the best thing



If operation failed, propagate the error  
and show it to the end user

End user retries if  
he/she wants to

let the user retry

Approach 2: Check and update

Idea: Get the status of payment

& process only when not already processed

Implementation

create a unique payment\_id  
↓  
and weave your API calls with it

Your API server can use

↗ this to get, check &  
update if needed

## Idempotent Flow

1. Payments service talks to Payment Gateway & generates a Payment ID
2. Payment service initiates the payment through this ID  
 $A \rightarrow B \quad \$20,000$
3. If payment service retries, it first checks the status of payment on ID  
if the status is COMPLETED then does not retry  
else retry through the same ID

