:years: 2015-2022

Idempotency

Idempotency is the way to make sure your specific action is only executed once. For example, if you have a button that is supposed to send a repayment, you don't want to repayment twice if the user clicks the button twice. Idempotency is a way to make sure that the action is only executed once.

There are two ways to use idempotency:

- HTTP Request with idempotency key header
- Batch request with batch item header

How it works

The `idempotency key` with `action name` and `entity name` is unique, and identify a specific command in the system. If no idempotency key is assigned to the request, the system will generate one for you.

1. User send a request
2. The system checks there are already executed commands with the same `idempotency key` and `action name` and `entity name`
3. The action based on the result of the check
    - If the request is completed the system return with the already generated result
    - If not completed, return HTTP 409 response
    - If the request is not completed, we process the requests and store the results in the database

Idempotency in HTTP requests

To achieve idempotency in HTTP requests, you can use the HTTP header from `fineract.idempotency-key-header-name` configuration variables (default `Idemptency-Key`). This header is a unique identifier for the request. If you send the same request twice, the second request will be ignored and the response from the first request will be returned.

Idempotency in Batch requests

In batch requests, you can set the idempotency key for every batch item, in the batch item `header` fields. The header key is from `fineract.idempotency-key-header-name` configuration variables (default `Idemptency-Key`).

Result of the request

- When the request is already executed and completed, the system will return a `x-served-from-cache` header with the value `true` in the response and return the original request body.
- When the request is already executed but still not completed, the system will return to HTTP 409 error code
- When the request is not executed, the system runs it normally and stores the result in the date