

Business date & Close of Business day concept in Fineract

Presented by Adam Saghy

FinTech Track



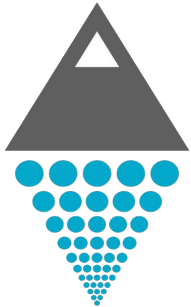
Halifax, NS
October 6-10, 2023

Who I am...

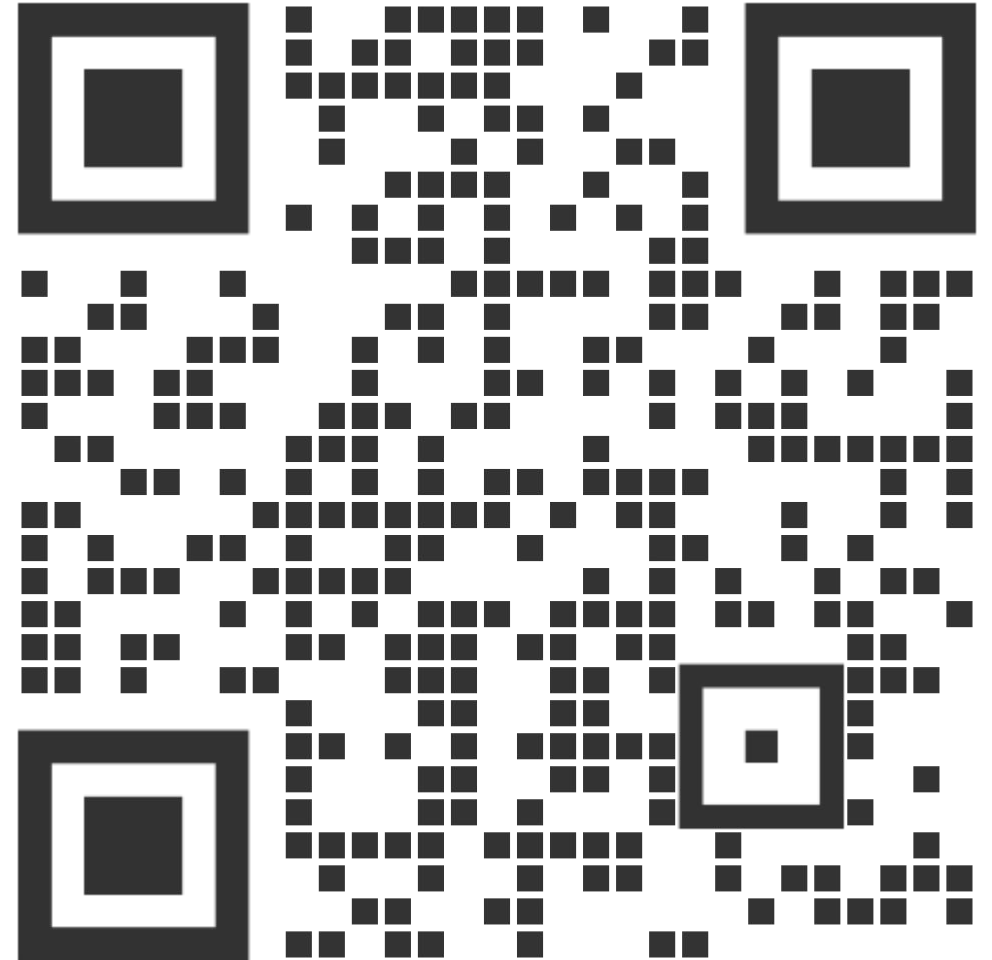


Adam Saghy – Principal Software Engineer &
Apache Fineract PMC member

adamsaghy@baasflow.com



To learn more about BaaSFlow and our
upstream contributions.
www.baasflow.com

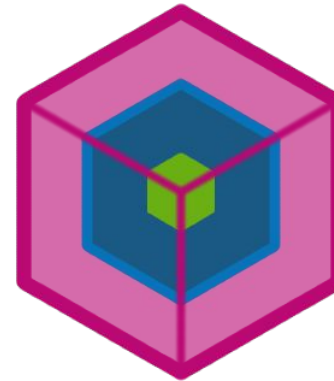


Business date & Close of Business day concept in Fineract



What is Fineract?

“Fineract is a mature platform with open APIs that provides a reliable, robust, and affordable core banking solution for financial institutions...”



Apache Fineract

<https://fineract.apache.org>

What is Fineract used for?

“For any organization, big or small, it provides the client data management, loan and savings portfolio management, integrated real time accounting, and social and financial reporting needed to bring digital financial services to a modern connected world”



Dates in Fineract



Created on
date

Executed on date

Submitted on
date

CoB date

Last modified on date

Transaction date

Tenant date

*Business
date*

Opened on date

Value date

Closed on date

Date definitions



Created date / Last modified

When the transaction was created / last modified (audit purposes). Date + time

Submitted on date / Posting date

When the transaction was posted. Tenant date or business date (depends on whether the logical date concept was introduced or not)

Transaction date / Value date

The date on which the transaction occurred or to be accounted for

Tenant date

Tenant time-zone date; its value is tied to the tenant calendar. Represents a day of the tenant time zone

Business date

Logical date; its value is not tied to the any time zone calendar. Represents a day of the logical / business calendar

Business day

Timeframe that logically group together actions on a particular business date

COB

Close of Business; concept of closing a business day

Cob date

Logical date; Represents the business date for actions during COB job execution

Date categories

System date

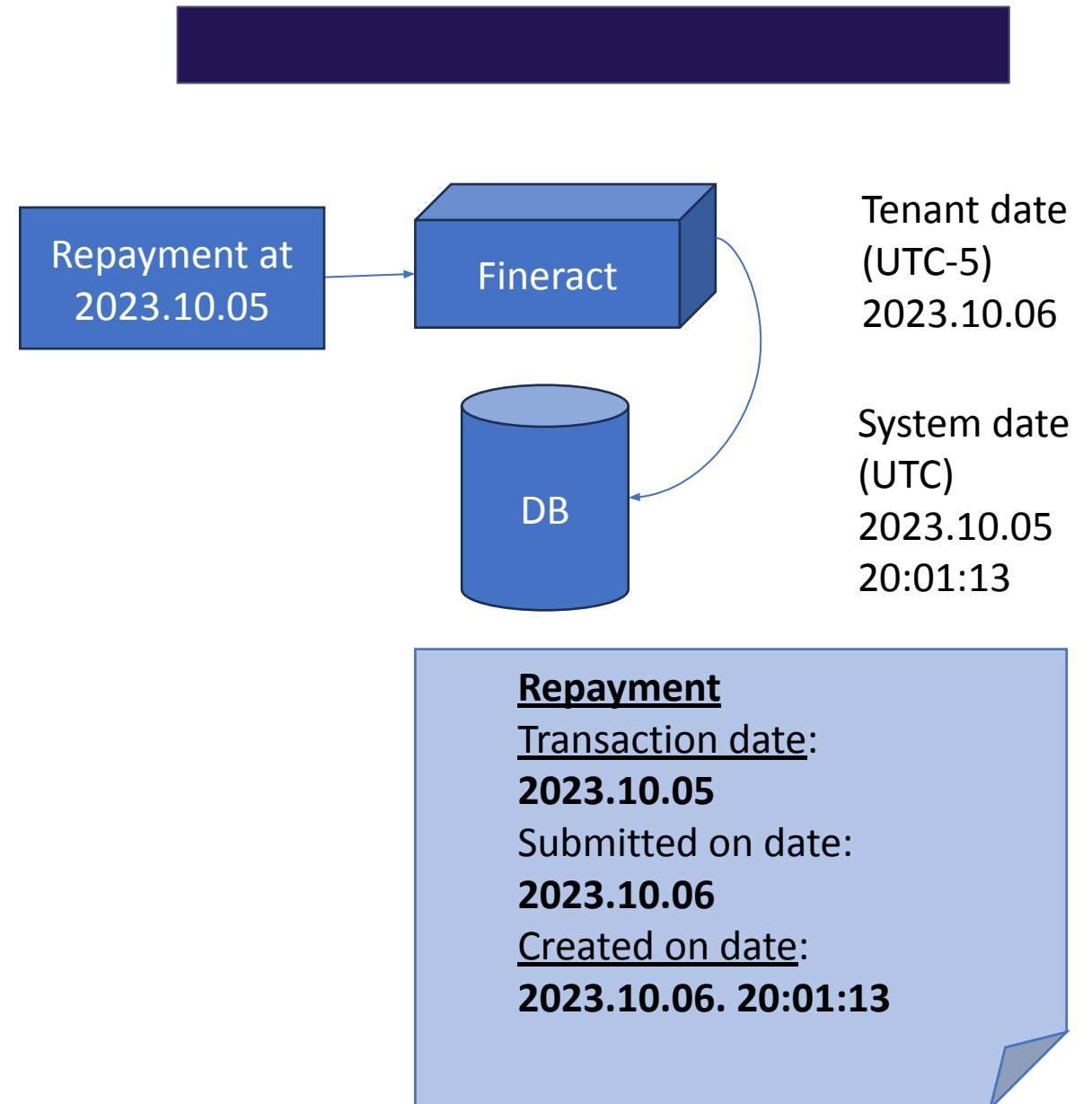
- For audit purposes
- UTC

Tenant date

- For business purposes
- Validations
- Tenant timezone

User-provided date

- For business purpose
- Backdated transactions
- Tenant timezone



Tenant date

Date based on the tenant time zone

- Tenant calendar

Tenant date tied to a time zone calendar

- Next day started at midnight

Pro

- Always be in sync

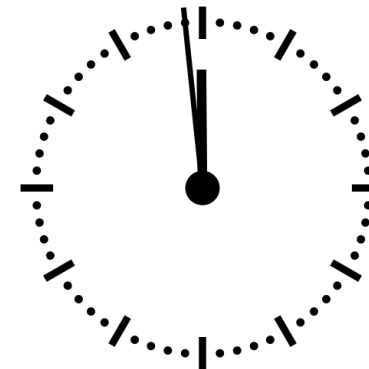
Con

- No control over the date
 - Cannot stop the time



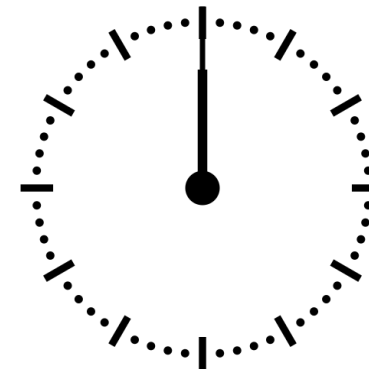
Time

Date



2023.10.10

23:59



2023.10.11

00:00

Problems with tenant date

The day end based on the tenant calendar

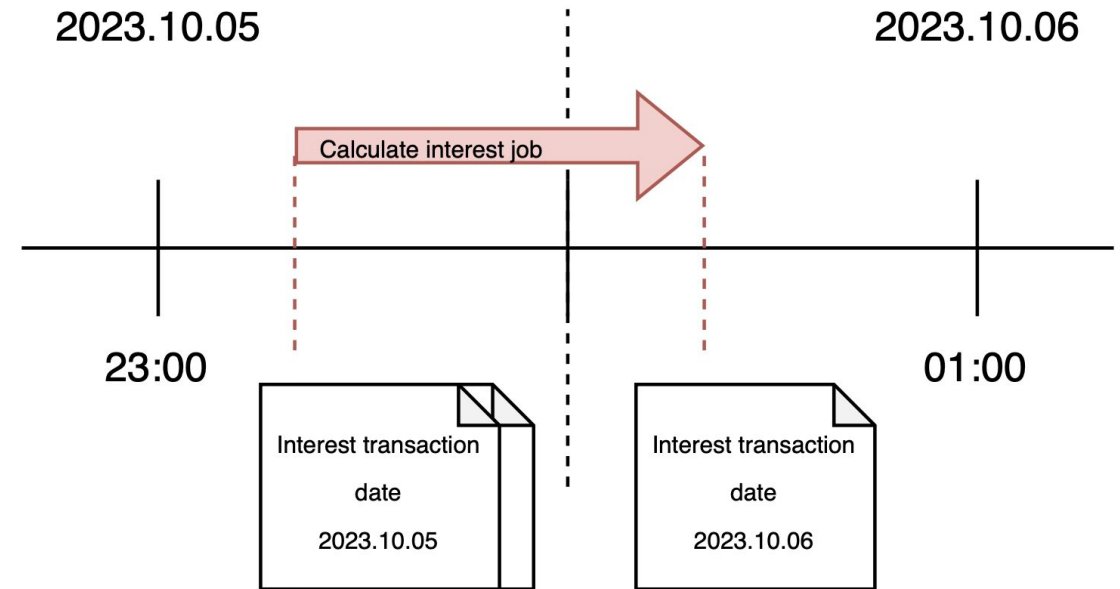
- Not following business needs sometimes

Execution of jobs either

- Before midnight
 - Might not finish before midnight
- On/After midnight
 - they are rather a start of day actions

Regulatory compliance

- Reports



Problems with tenant date



Lack of flexibility

- Limited disaster recovery
- Limited support for testing complex, multi-day use cases

No real "Close of business day" support

- Calculate and add interest / penalty transaction as part of closing the actual day
- No particular point where new transactions are accounted to next day, but as part of closing the actual day the jobs which calculates and creates interest / penalty still accounted to the day we are "closing"

Close of Business day concept



Having a point from where new, incoming actions belong to the next day (existed)

- If tenant date was used, this point was midnight

The requests which were initiated on a particular business day the outcome must belong to that particular day (new)

- Same date to be used for all related actions

Having “Close of Business day” relevant jobs (new)

- This jobs are using the “business day” as the relevant date and the outcome of all actions still belong to this date

Executing CoB on business entity before accepting incoming “next day” actions (new)

- Data consistency

Regulatory Compliance (new)

- The business date is crucial for regulatory reporting and compliance. Financial institutions must adhere to regulations related to how they calculate and record transactions, making the business date a vital component of compliance.

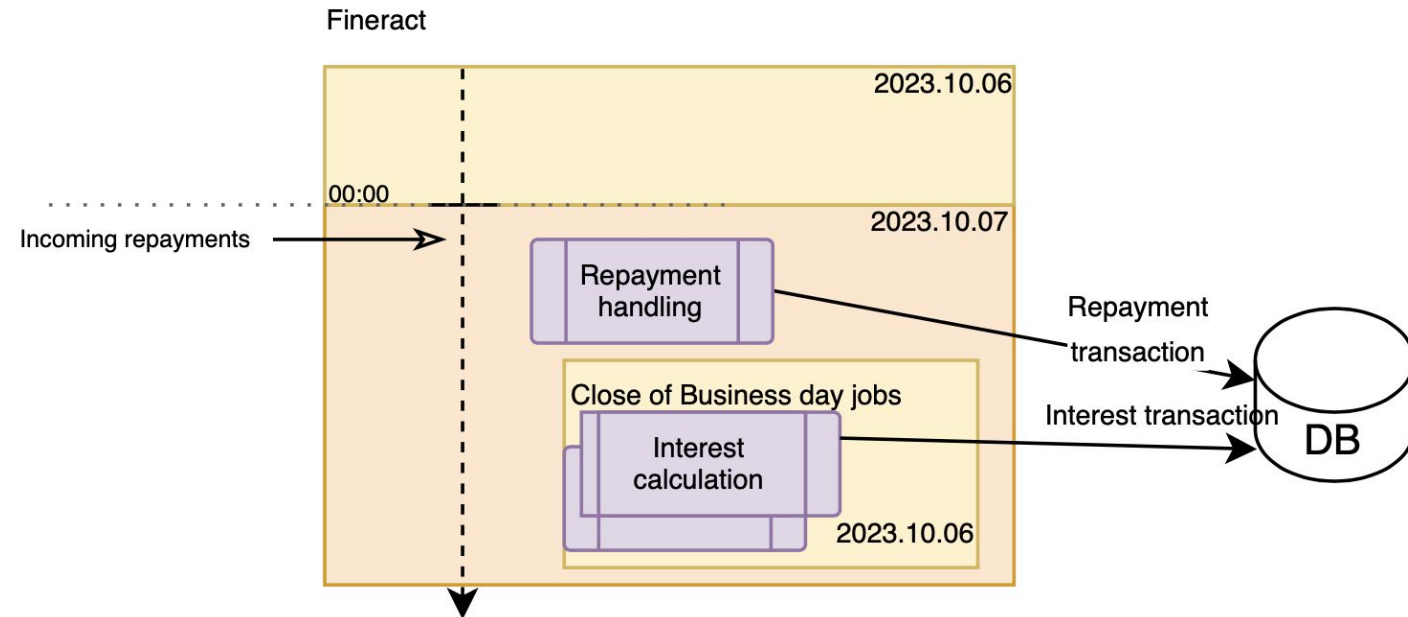
What could be the solution?

We need at least 2 new dates

- Business date
- Close of business date

We need at least 2 contexts

- Real-time actions
 - Incoming transactions
 - Close of (business) day actions
 - Internal (job triggered) transactions



Business date

Date based on logical calendar

- Business calendar

Business date can differ from tenant time zone date

- Next day can be started any time

Pro

- Date follows the business needs

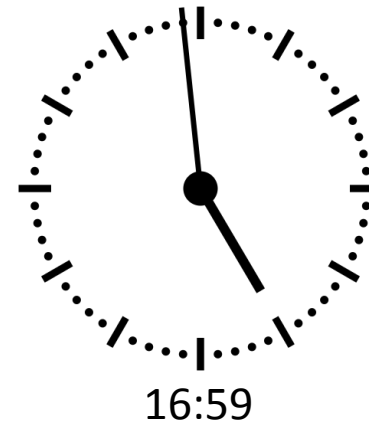
Con

- No auto sync

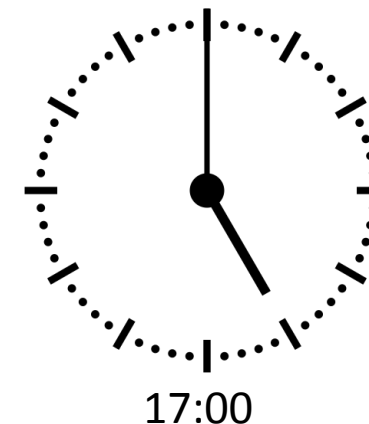


Time

Date



2023.10.10



2023.10.11

Business date



It does not support time parts

It can be managed manually (via API call) or automatically (via scheduled job)

All business actions during the business day shall use this date:

- Posting / submitted on date of transactions
 - Submitted on date of actions
 - (Regular) jobs

It will be used in every situation where the transaction date / value date is not provided by the user

If provided then the user provided date shall be validated:

- Disbursal date
- Posting/Submitted date
- Reversal date

Will not be used for audit purposes:

- Created on date
- Updated on date

Business day

Timeframe that logically group together actions on a particular business date

- Accounts, transactions, events...

The length of the business day can vary like

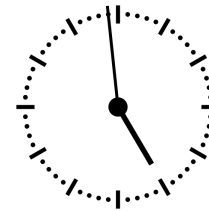
- On week days till 16:59
- On weekends till 16:59 of Monday

Support

- Disaster recovery
- Testing
- Close of Business day

Week days

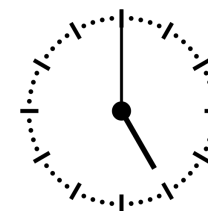
2023.10.05.



2023.10.05. 16:59



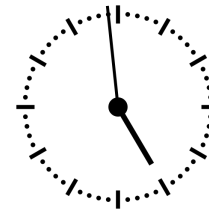
2023.10.06.



2023.10.06. 17:00

Weekend days

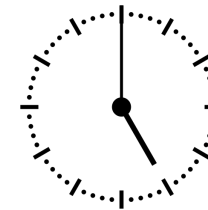
2023.10.06.



2023.10.06 16:59



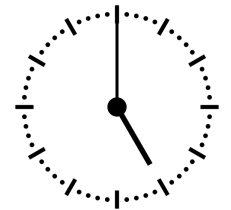
2023.10.09.



2023.10.07 17:00



2023.10.09.



2023.10.08 17:00

CoB date

Closing a business day date

- Based on the business calendar

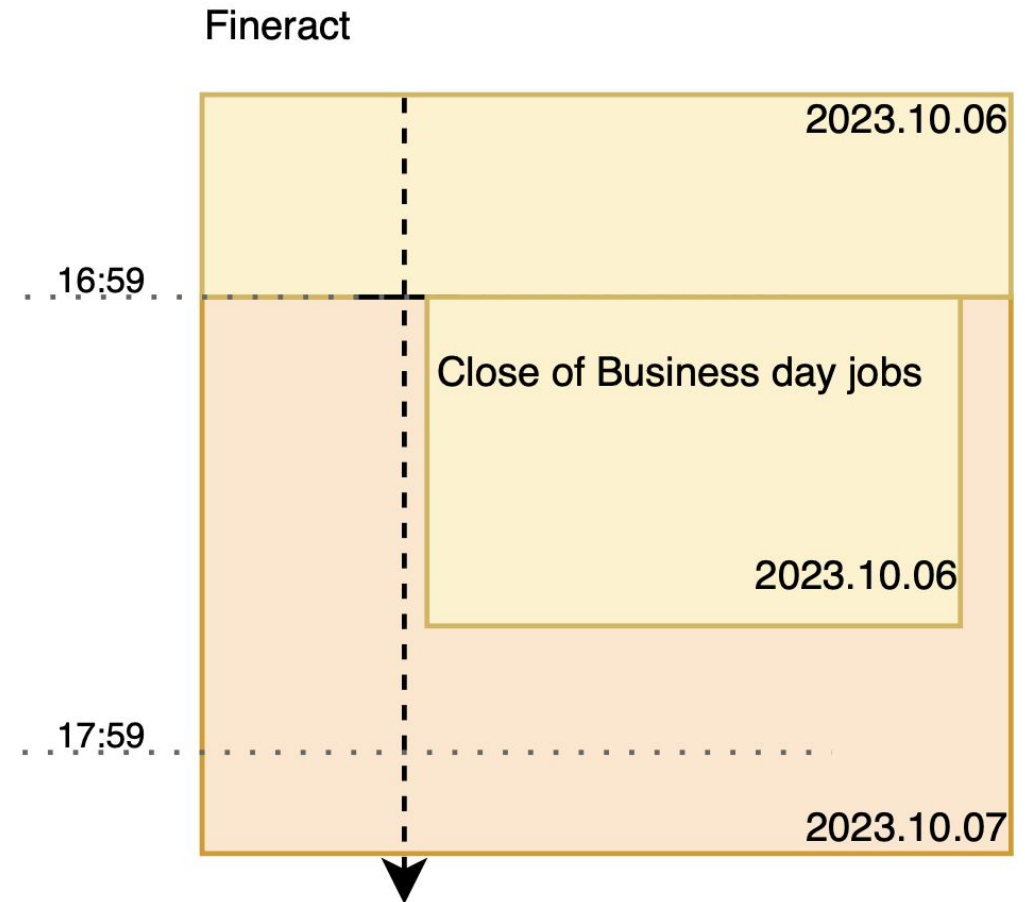
During the CoB jobs this date to be used as the business date

Usually it is in sync with business date (configurable)

- Business date minus 1 day

Pro

- During the execution of CoB jobs the actions and transactions are still belongs to the business day



How to configure in Fineract?



- Enable “business date”
 - Global configuration
- Set whether the “business date” and “CoB date” should be kept in sync
- Set the actual “business date” via API
- Update the “business date” when ever the business day ends
 - via API
 - via Job

Business date data model

It stores the business date and the cob date

It can be easily extended with additional columns (for future use cases)

Type: Define the type of the business date

Values:

- BUSINESS_DATE
- COB_DATE

Value: Store the date value

Values:

- DATE without time and time zone
- Example: "2022-05-06"

Table

```
m_business_date {  
    "id": SERIAL,  
    "type": STRING,  
    "value": DATE,  
    // audit fields and  
    versioning  
}
```

Constraints

- Type must be UNIQUE!
- Must be initialised before first use!
- Value cannot be empty or null!
- By default it goes with 2 entries:
 - Business date
 - COB date
- Date format is "YYYY-MM-DD"

Business date configuration



Two new global configuration entries were introduced

Enable business date (default value: false)

When it is enabled, the business date will be used instead of the tenant date

When it is not enabled, the business date logics are falling back to use the tenant date.

Enable auto CoB date adjustment (default value: true)

When it is enabled, the CoB date will be automatically recalculated based on the business date

Formula: COB date = business date - 1 day

When it is not enabled, the cob date will not be automatically recalculated hence need to update manually

Business date actions via API



To manually adjust the business date and/or cob date

Set a date

POST /api/v1/businessdate

Request

```
{
  "type": "BUSINESS_DATE",
  "date": "2022.05.13",
  "dateFormat": "yyyy.MM.dd"
}
```

Response

```
{
  "changes": {
    "BUSINESS_DATE": [2022, 5, 13],
  }
}
```

Business date actions via API



To fetch business dates

To fetch business a particular date

Fetch dates

Fetch a particular date

GET /api/v1/businessdate

GET /api/v1/businessdate/BUSINESS_DATE

Response

Response

```
[
  {
    "type": "BUSINESS_DATE",
    "date": [2022, 5, 13]
  },
  {
    "type": "COB_DATE",
    "date": [2022, 5, 12]
  }
]
```

```
{
  "description": "Business Date",
  "type": "BUSINESS_DATE",
  "date": [2023, 3, 2]
}
```

Business date actions via Job



Alternatively we can increase the business date/cob date by 1 day.

Execute “Increase Business date by 1 day” job

```
POST /api/v1/jobs/32?command=executeJob
```

Pseudo code

```
{
  IF (ENABLE BUSINESS_DATE == TRUE) {
    //VALIDATION
    IF (ACTUAL_BUSINESS_DATE == null) {
      THROW ERROR;
    }
    IF (ENABLE AUTOMATIC COB_DATE ADJUSTMENT == TRUE) {
      NEW_COB_DATE = ACTUAL_BUSINESS_DATE;
    }
    NEW_BUSINESS_DATE = ACTUAL_BUSINESS_DATE + 1 DAY;
  }
}
```

Contextual dates



“The requests which were initiated on a particular business day the outcome must belong to that particular day”

- Introduced contextual dates which does not got changed during the processing of the request

ThreadLocalContext

- Fetch from database and store the business date
- All business logic shall fetch the business date from this context
- We had to distinguish at least two situation:
 - Default context
 - By default context
 - COB job context
 - Data during the COB job execution

Contextual dates

Based on which context we are in, the logical date should be either the business date or the cob date

Per request basis the business dates to be fetched from the DB

Fetching from cache is an option, but must be careful with it!

Initializing the business context must be one of the first steps

Asynchronous event handlers must reinitialize in the ThreadLocalContextUtil

Refactor functionalities to retrieve the business date from ThreadLocalContextUtil rather than using the tenant date

Pseudo class

```
Class ThreadLocalContextUtil {  
    //fields  
    Date businessDate;  
    Date cobJobDate;  
    Boolean isCobContext;  
  
    //methods  
    Date getBusinessDate() {  
        IF (isCobContext == TRUE){  
            RETURN cobJobDate;  
        } ELSE {  
            RETURN businessDate;  
        }  
    }  
}
```

Changes in Fineract



Modify at all the relevant places where the tenant date was used:

- With very limited exceptions all places where the tenant date is used we need to modify to use the business date

Replace system date with tenant date or business date (exceptions may apply)

Review value date (transaction date) and posting date (submitted on date), created on date usages

Jobs to be reviewed and enhanced

- Majorly focusing on COB

Native queries to be reviewed and enhanced

Reports to be checked and enhanced

Every table where update is supported the `AbstractAuditableWithUTCDateTimeCustom` should be implemented

Enhance transactions and journal entries date handling to fit for business date concept

Use case



Apply for a loan

Tenant date: 2022-05-23 14:22:12

Business date: 2022-05-22

Submitted on date: 2022-05-23

Outcome: **FAIL**

Message: **The date on which a loan is submitted cannot be in the future.**

Reason: Even the tenant date is 2022-05-23, but the business date was 2022-05-22 which means anything further that date must be considered as a future date.

Q & A

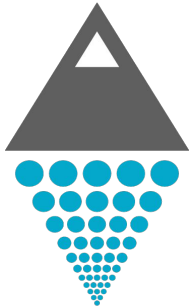


Thank you!



Adam Saghy

adamsaghy@baasflow.com



To learn more about BaaSFlow and our
upstream contributions.
www.baasflow.com

