# CG Account Ledger

Contents

[CG Account Ledger 1](#_Toc166831737)

[Architecture: 1](#_Toc166831738)

[Software/Tools: 2](#_Toc166831739)

[Account Open Event (Command Gateway) 2](#_Toc166831740)

[Account Status Change API (Command Gateway) 3](#_Toc166831741)

[Asset Posting API (Command Gateway) 3](#_Toc166831742)

[Update Posting API (Command Gateway) 5](#_Toc166831743)

[Get Wallet Balance API (Query Gateway) 6](#_Toc166831744)

[Event Exception 6](#_Toc166831745)

[Testing : 7](#_Toc166831746)

[AXON Health Check 7](#_Toc166831747)

[H2 Database health Check 8](#_Toc166831748)

[Account Status Change API 9](#_Toc166831749)

[Asset Posting API (Move asset from wallet to another wallet for same account) 11](#_Toc166831750)

[Update Posting Status 14](#_Toc166831751)

[Get Wallet Balance 15](#_Toc166831752)

## Architecture:

Cg Account Ledger has implemented CQRS(Command Query Responsibility Segregation) design pattern to support the following API as an Event Driven Architecture.

* Account Open Event
* Account Status Change API
* Asset Posting API (Move one asse from one wallet to another)
* Update Status of Posting
* Get Wallet Balance

## Software/Tools:

**Technology**: Java 17, Spring Boot 3.2.5

**Feature**: Axon Server (Store Events to support CQRS)

**Tools**: Spring Tool Suite, Postman

**Database**: In Memory H2 database

## Account Open Event (Command Gateway)

CG Account ledger will open 5 different accounts with multiple wallets and save it in h2 database whenever the application starts. Following is the flow of the event follows CARS

**AccountAggregate**

send

Command Gateway

start

Aggregate

Command Line Runner

Spring Boot App

Handle Event

Publish event

**AccountOpenedEvent**

**OpenAccountCommand**

**AccountOpenedEventHandler**

save

**AccountRepository**

H2 DB

Event Store

Listen event

Repository

Event Handler

CG

* Command Line Runner get initialized once Spring Boot Application Starts.
* Account Open Request converts to Open Account Command in command line runner.
* Once command is created, Command request sent to AccountAggreagte .
* AccountAggreagte will handle the command request and create an event after implementing/creating account logic.
* AccountAggreagte will publish AccountOpenedEvent to Event Store.
* AccountOpenedEventHandler will listen the event and send the request to repository to save the opened account details in H2 database.

## Account Status Change API (Command Gateway)

CG Account Ledger accepts the request to update the status of the account i.e. OPEN to CLOSED. Following is the flow of the API.

**AccountAggregate**

/api/accounts /{accountId}/status

Controller

API

send

Command Gateway

Aggregate

PATCH

Handle Event

Publish event

**AccountStatusChangeEvent**

**AccountStatusCommand**

**AccountOpenedEventHandler**

save

**AccountRepository**

H2 DB

Event Store

Listen event

Repository

Event Handler

CG

* AccountStatusController receives the request to update the status of the account, Request has converted to Command (AccountStatusCommand) in controller
* AccountIterceptor has been called before sending the command to aggregate for the business validation (account id not found).
* Command has sent to Aggregate to create an even and publish it, once validation is completed.
* AcountAggregate receives AccountStatusCommand and create AccountStatusChangeEvent and publish.
* AccountOpenedEventHandler listen to AccountStatusChangeEvent and send the request to repository to update the status
* update the status

## Asset Posting API (Command Gateway)

CG Account Ledger accepts the request to move the asset from one wallet to another wallet i.e. move fund from HKD wallet to USD wallet only in a same account. Following is the flow of the API.

**AssetPostingAggregate**

/api/accounts/{accountId}/posting

Controller

API

send

Command Gateway

Aggregate

POST

Handle Event

**AssetPostingEvent**

Publish event

**AssetPostingCommand**

**AssetPostingEventHandler**

**AssetPostingRepository**

save

H2 DB

Event Store

Listen event

Repository

Event Handler

CG

* AssetPostingController receives the request to move the asset from one wallet to another wallet, Request has converted to Command (AssetPostingCommand) in controller
* PostingInterceptor has been called before sending the command to aggregate for the business validation (wallet id not found etc.).
* Command has sent to Aggregate to create an even and publish it, once validation is completed.
* AssetPostingAggreagte receives AssetPostingCommand and create AssetPostingEvent and publish.
* AssetPostingEventHandler listen to AssetPostingEvent and send the request to repository to update the status

## Update Posting API (Command Gateway)

CG Account Ledger accepts the request to move the asset from one wallet to another wallet i.e. move fund from HKD wallet to USD wallet only in a same account. Following is the flow of the API.

**AssetPostingAggregate**

/api/accounts/{accountId}/posting

Controller

API

send

Command Gateway

Aggregate

PUT

Handle Event

**AssetPostingStatusChangeEvent**

**AssetPostingStatusChangeCommand**

Publish event

**AssetPostingEventHandler**

**AssetPostingRepository**

save

H2 DB

Event Store

Listen event

Repository

Event Handler

CG

* AssetPostingController receives the request to update the posting status i.e. PENDING to CLEARED, Request has converted to Command (AssetPostingStatusChangeCommand) in controller
* PostingInterceptor has been called before sending the command to aggregate for the business validation (wallet id not found etc.).
* Command has sent to Aggregate to create an even and publish it, once validation is completed.
* AssetPostingAggreagte receives AssetPostingStatusChangeCommand and create AssetPostingStatusChangeEvent and publish.
* AssetPostingEventHandler listen to AssetPostingStatusChangeEvent and send the request to repository to update the status

## Get Wallet Balance API (Query Gateway)

CG Account Ledger accepts the request to give the balance of the wallet. Following is the flow of the API.

**WalletBalanceProjection**

/api/accounts/{accountId}/balance

Controller

API

send

Query Gateway

Projection

Get

Handle Event

**WalletRepository**

**GetWalletBalanceQuery**

Fetch the record

H2 DB

Repository

* WalletBalanceQueryController receives the request to provide the balance of the particular wallet, Request has converted to Query (GetWalletBalanceQuery) in controller
* Query Request has send to Projection/Query handler WalletBalanceProjection. Wallet Projection will make the call to h2 database to fetch the balance details for the wallet.

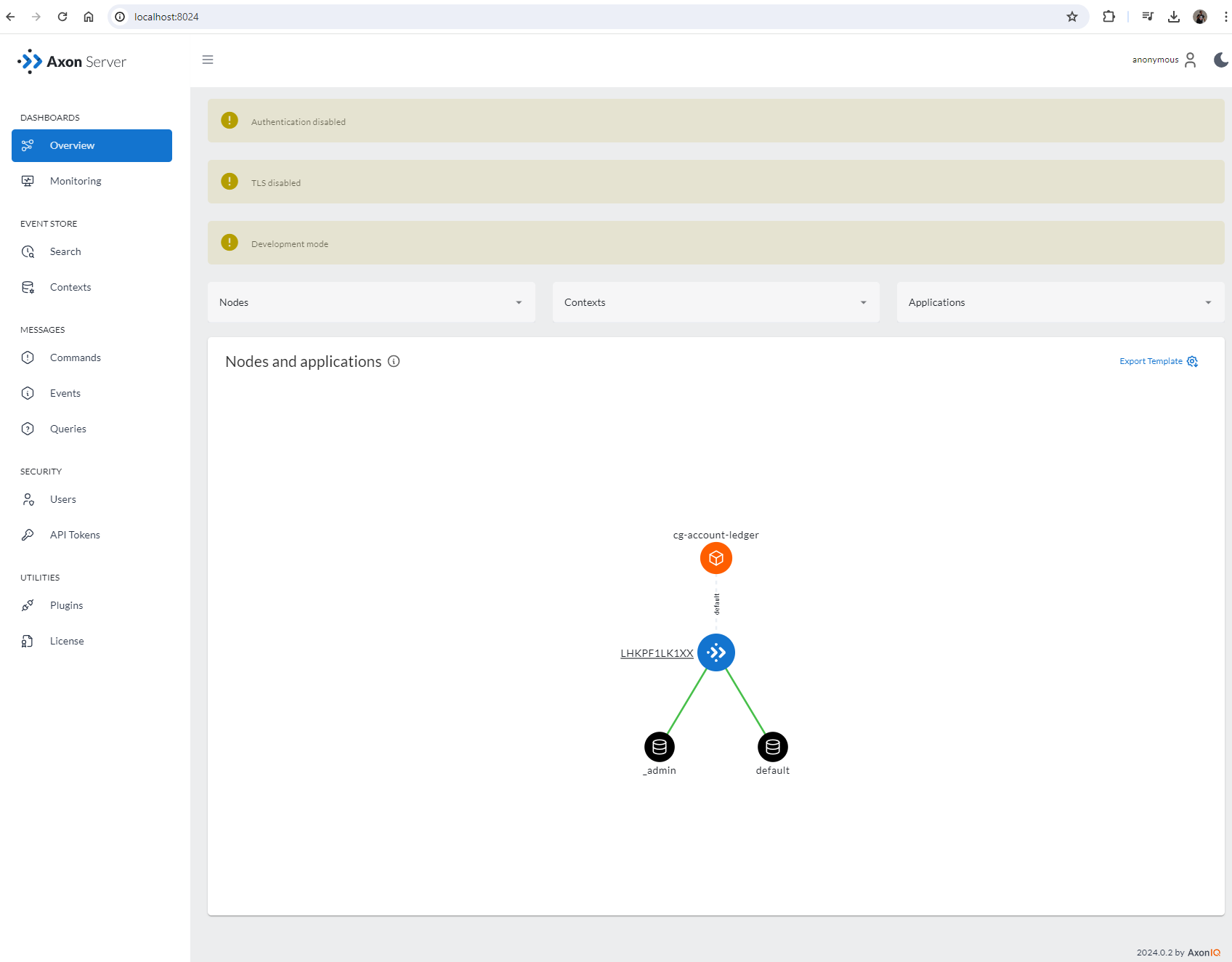
## Event Exception

Event error Listener( InvocationErrorHandler) has been registered in the cg account ledger application to revert the process if nay exception comes while processing the events

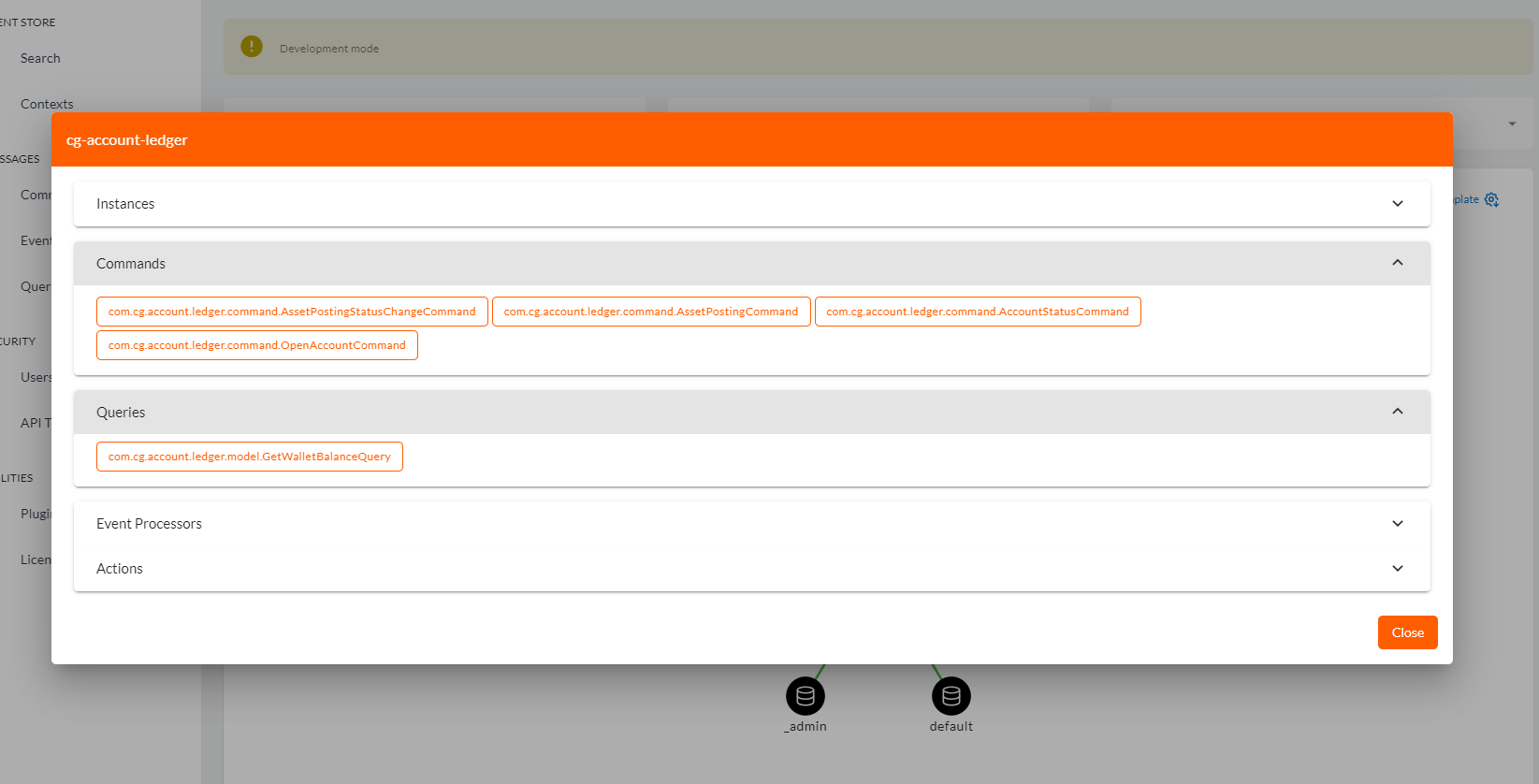
## Testing :

### AXON Health Check

Application has registered with AXON

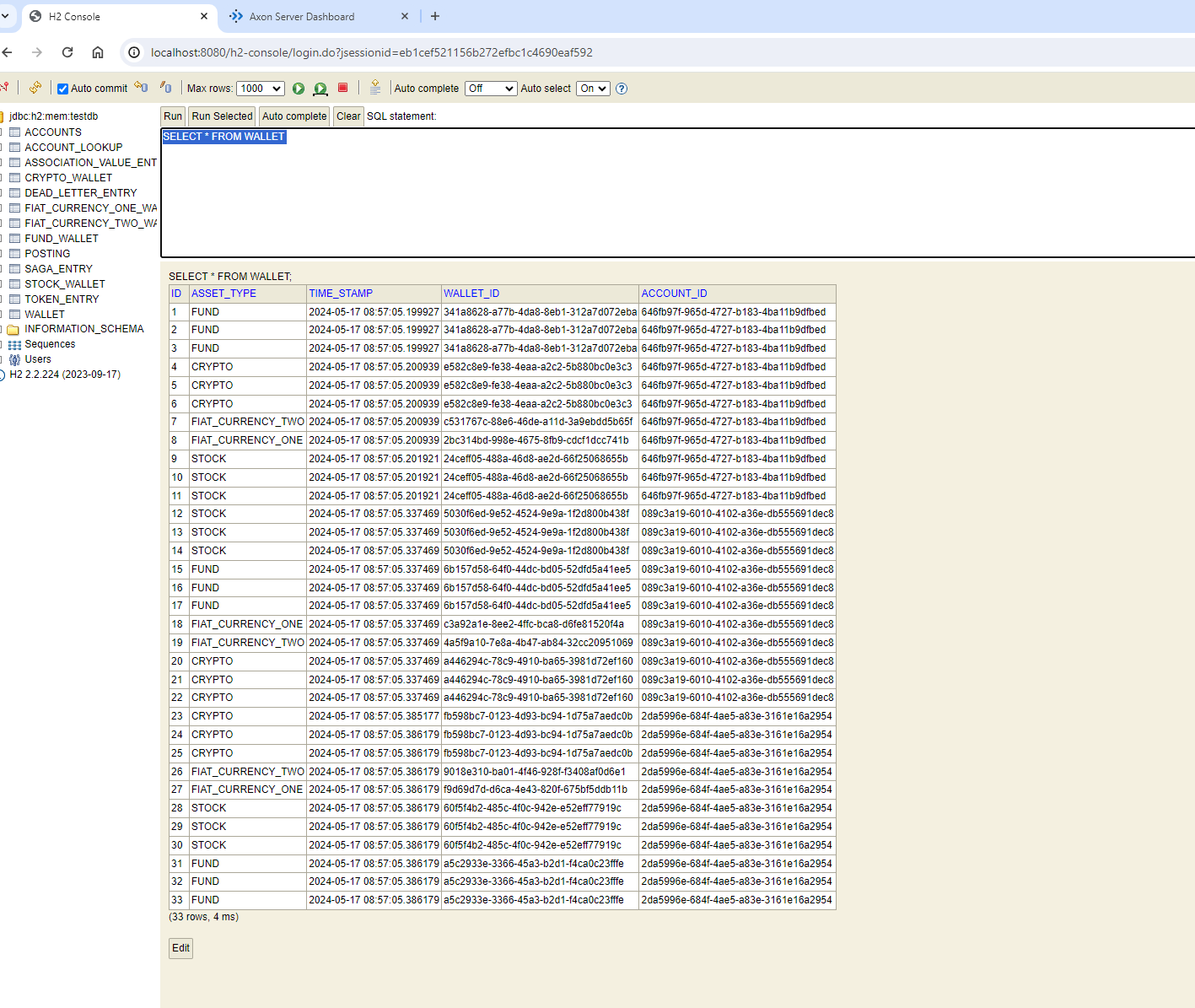


All the commands and Query registered with AXON



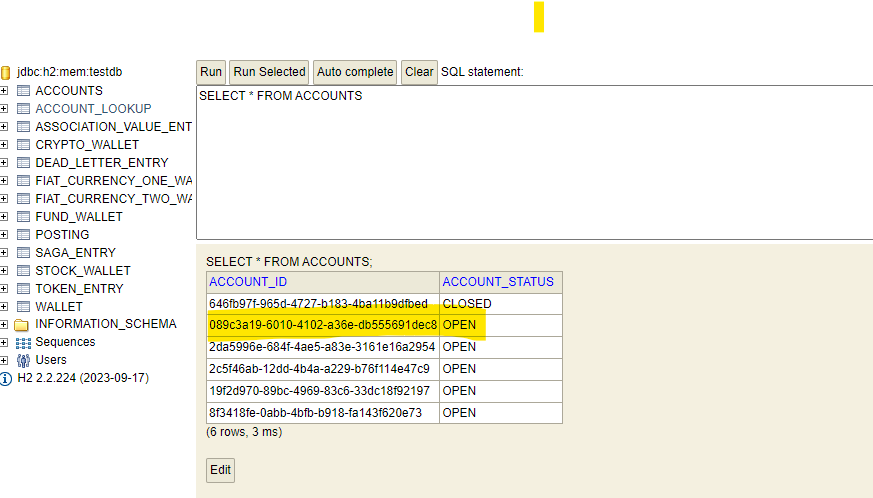
### H2 Database health Check

All the database tables got created, Open Account Command Created initial account records on database

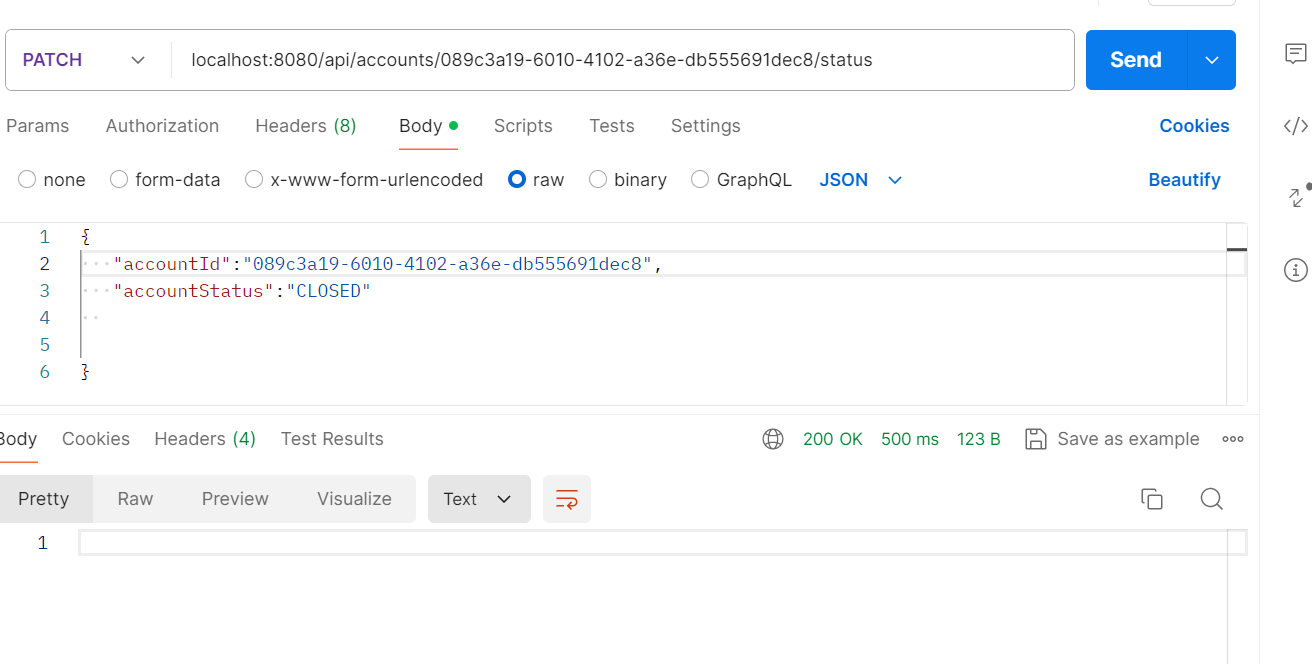


Account Status Change API

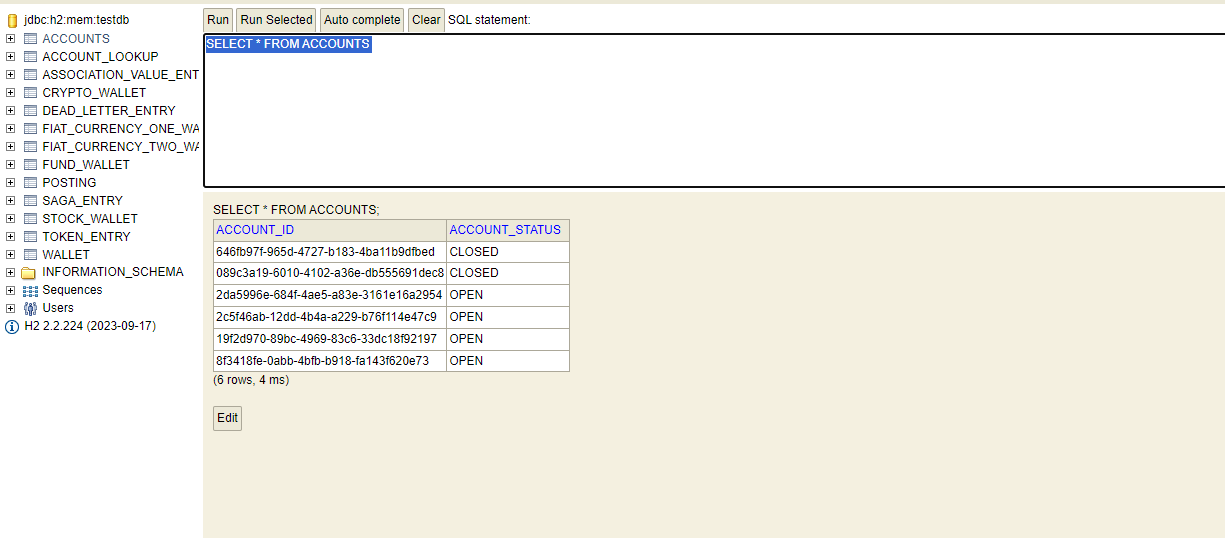
I will update the account status for the account id : **089c3a19-6010-4102-a36e-db555691dec8** from open to close



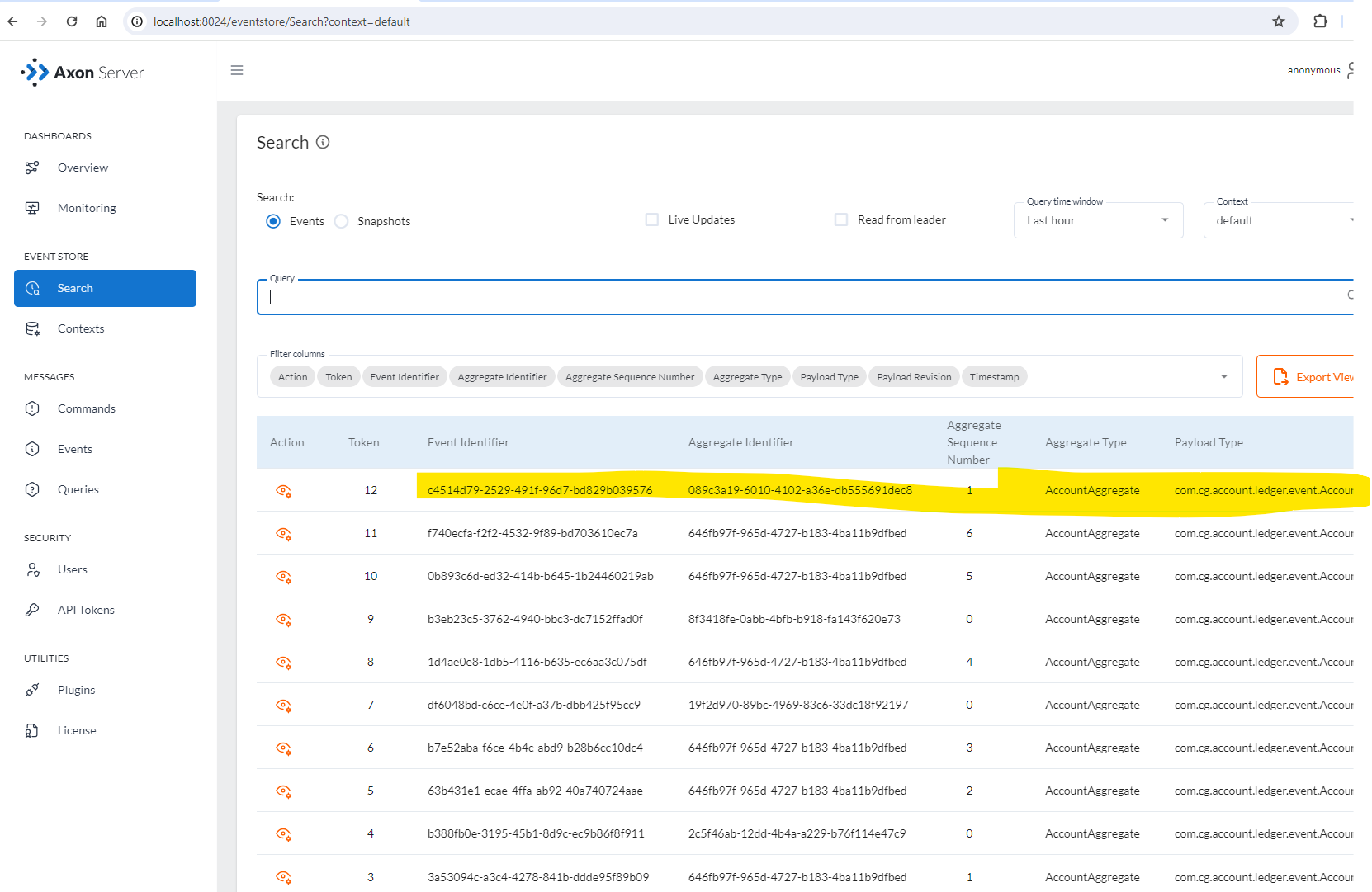
API call from Postman



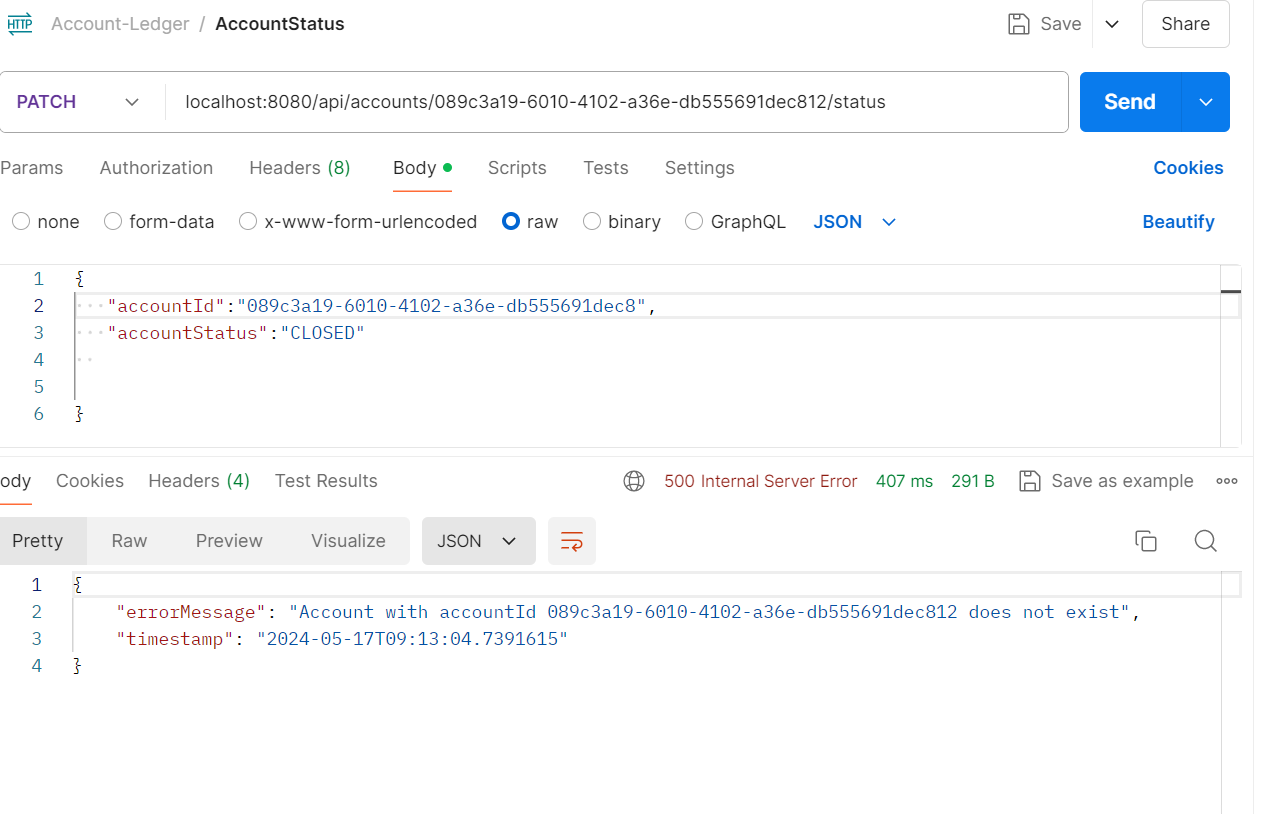
Record Updated



Event on Axon UI

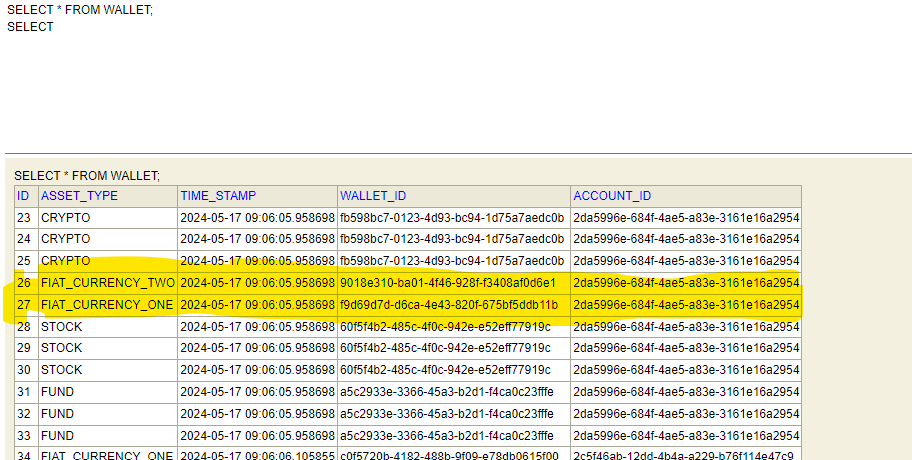


Negative Use Case (Account Not Found)

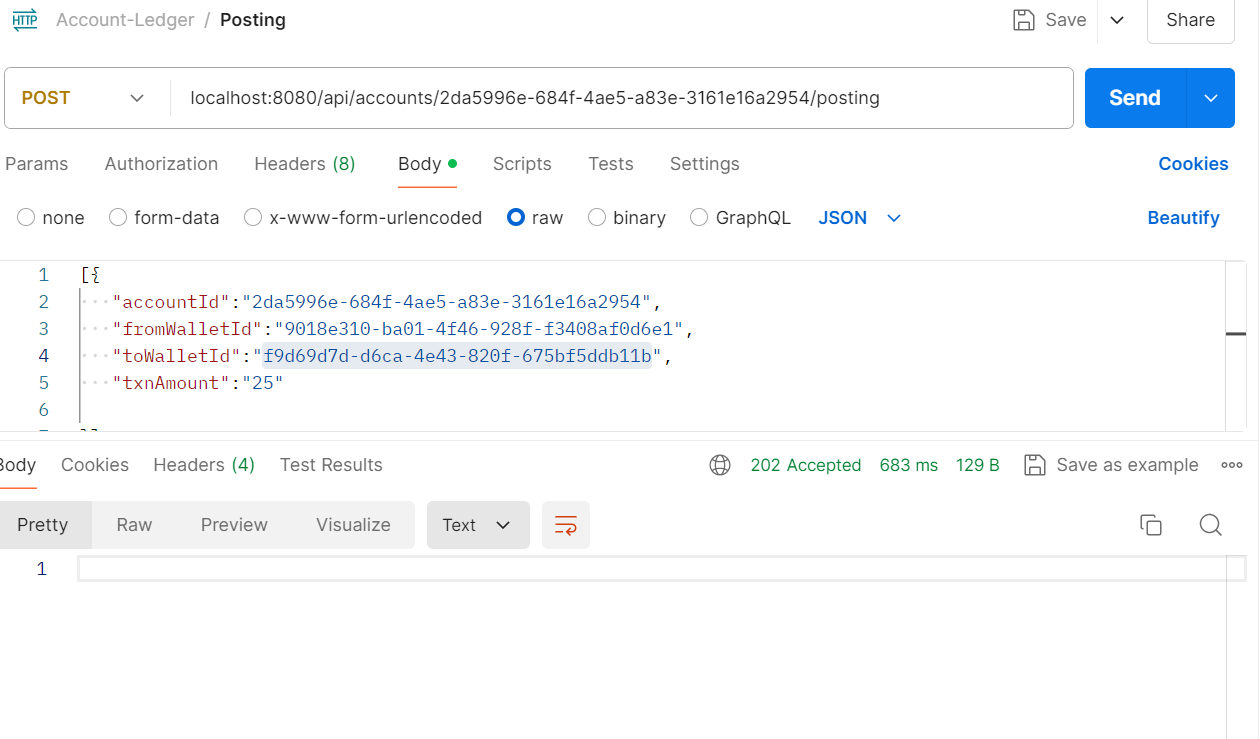


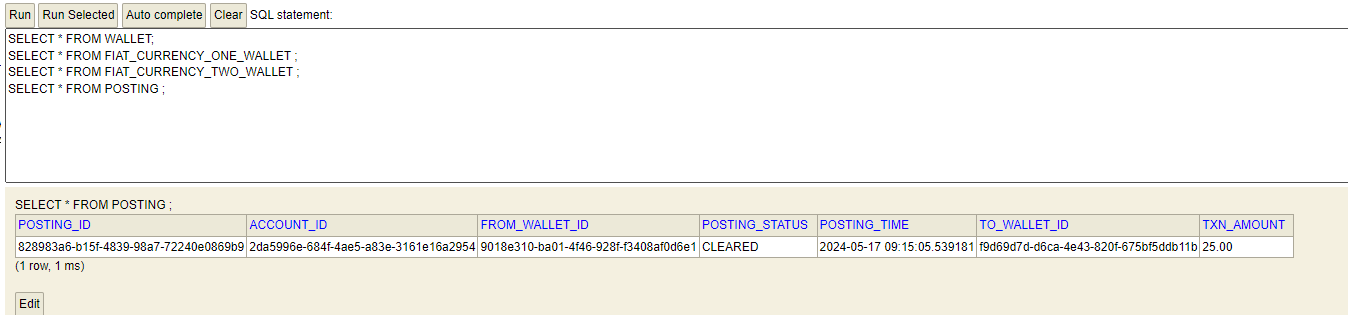
### Asset Posting API (Move asset from wallet to another wallet for same account)

I’ll move asset from wallet “**9018e310-ba01-4f46-928f-f3408af0d6e1**” to wallet “**f9d69d7d-d6ca-4e43-820f-675bf5ddb11b**” for account “**2da5996e-684f-4ae5-a83e-3161e16a2954**” and transaction amount is 25.

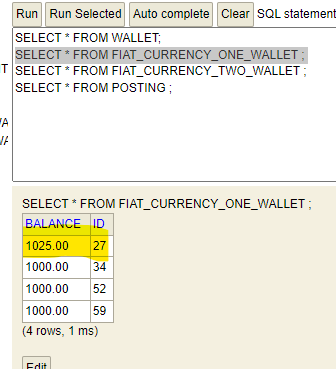


API call from postman

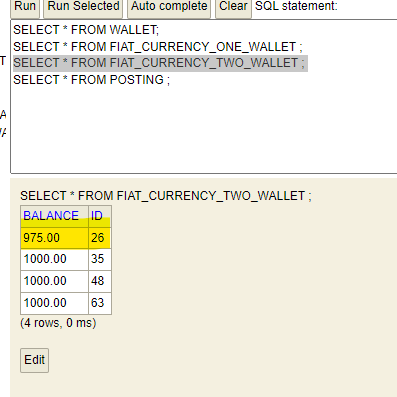




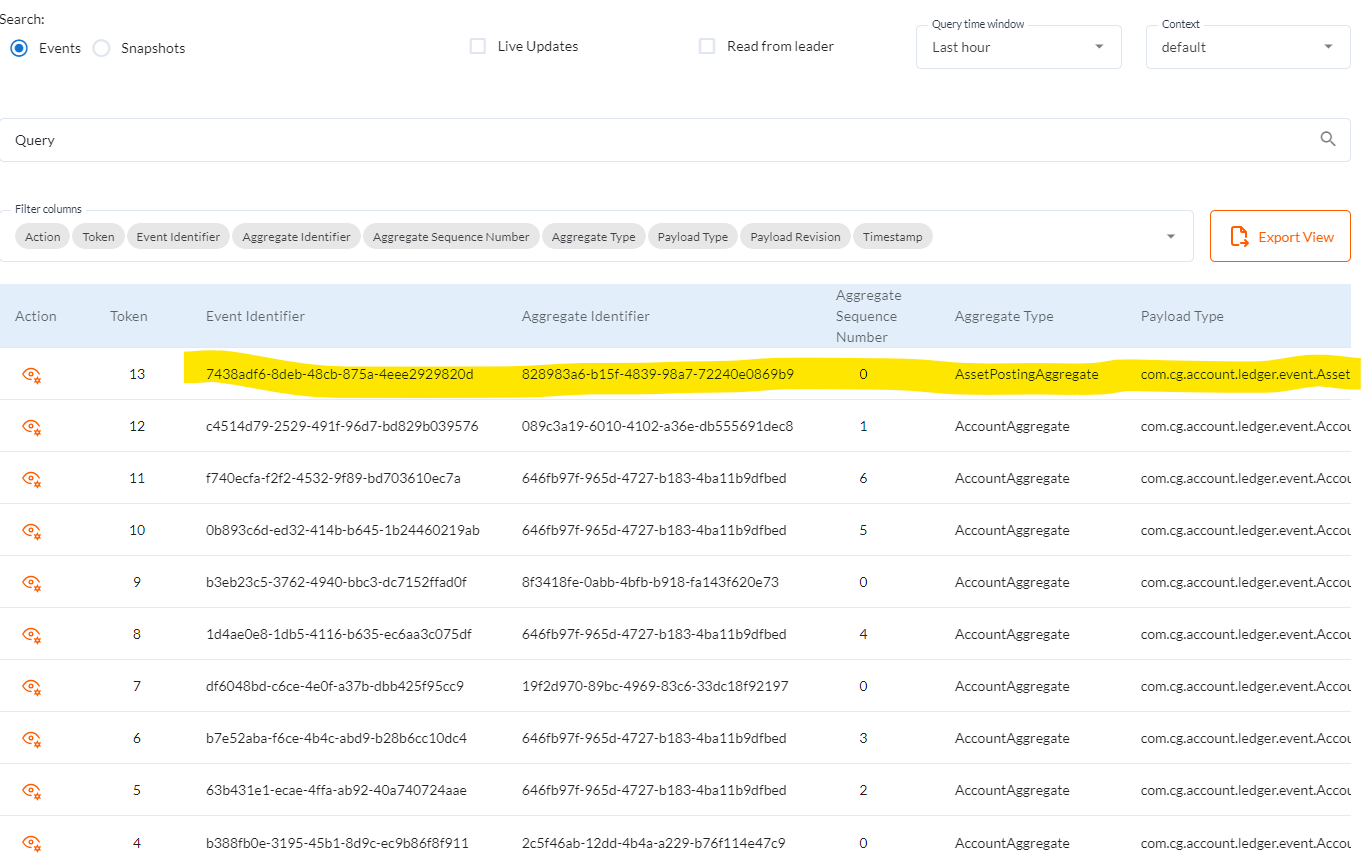
To wallet (**f9d69d7d-d6ca-4e43-820f-675bf5ddb11b)**



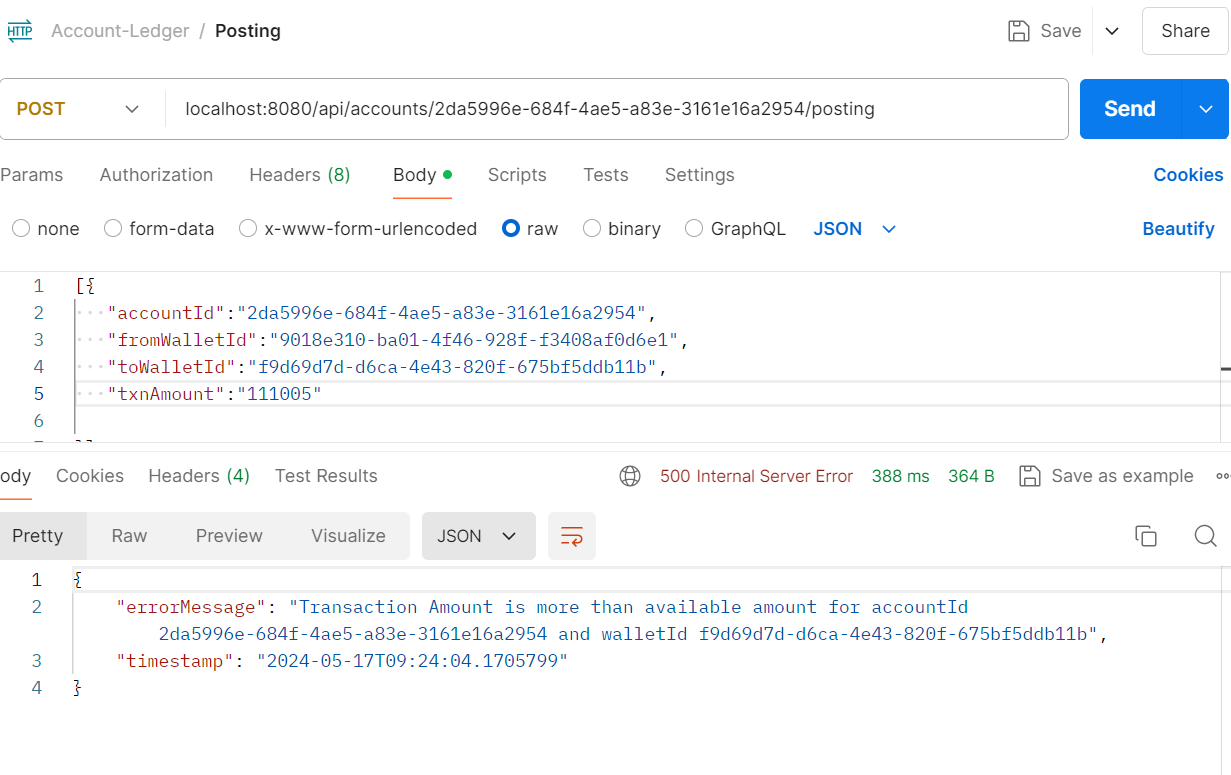
From wallet (**9018e310-ba01-4f46-928f-f3408af0d6e1**)



Axon Event

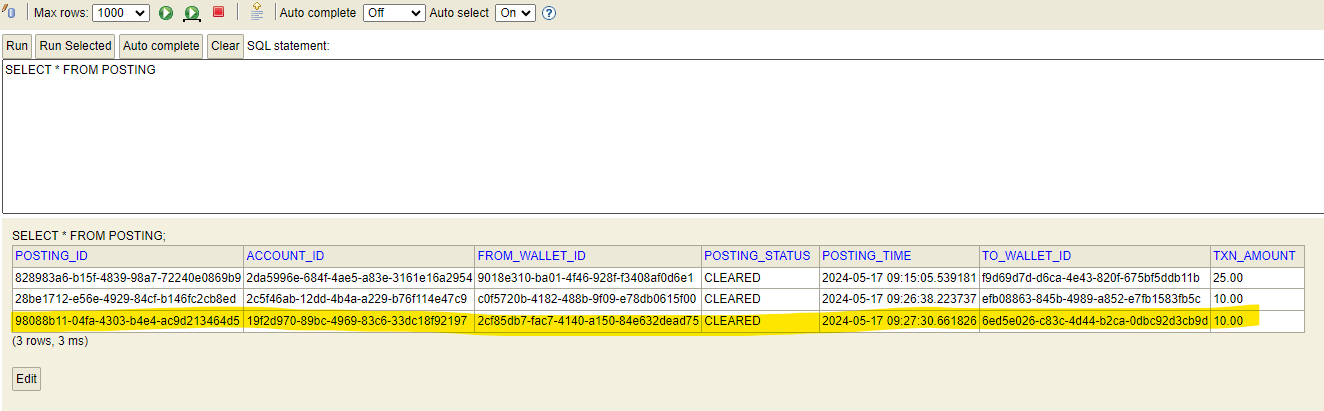


Negative Use Case

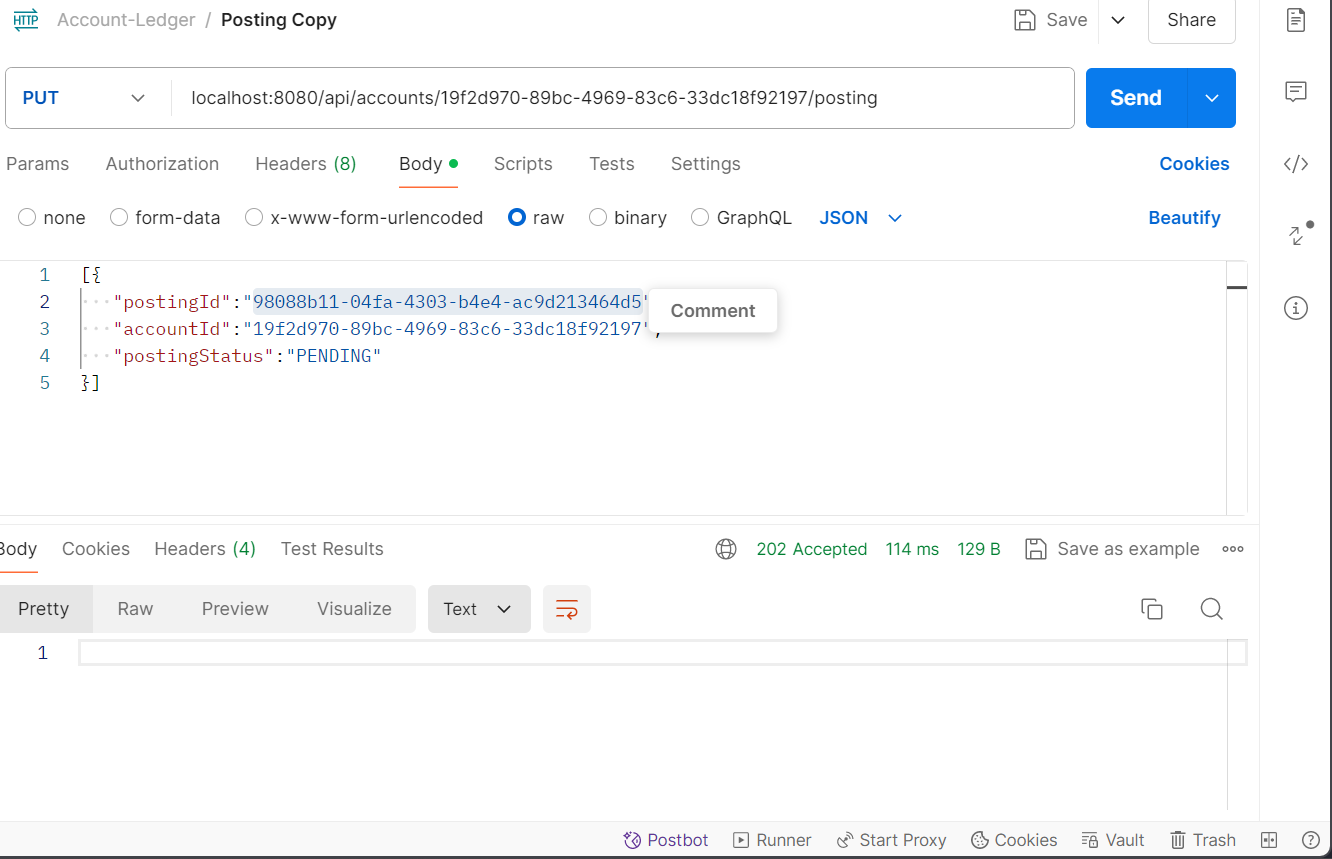


### Update Posting Status

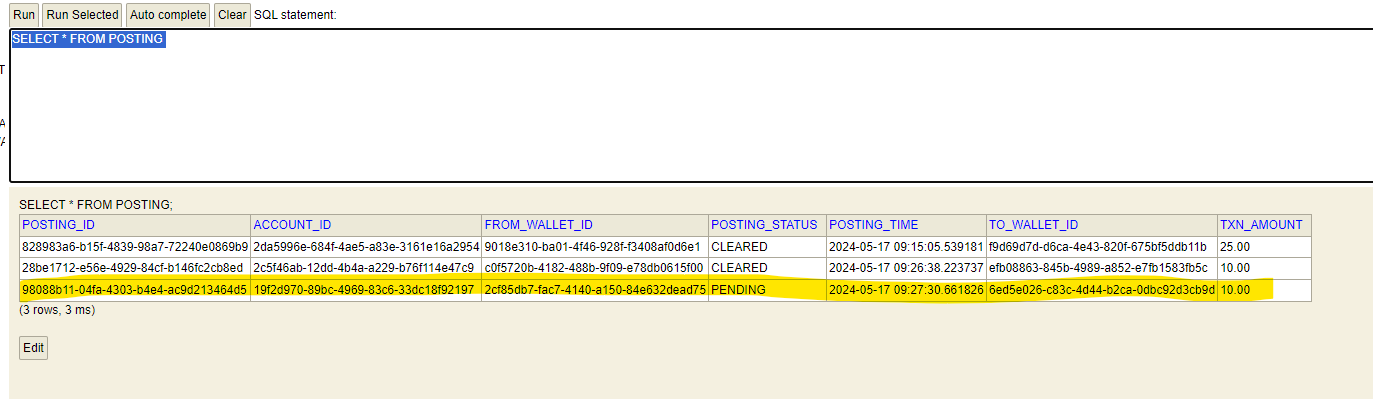
I will update the Posting status to “PENDING” of posting id “**98088b11-04fa-4303-b4e4-ac9d213464d5**”



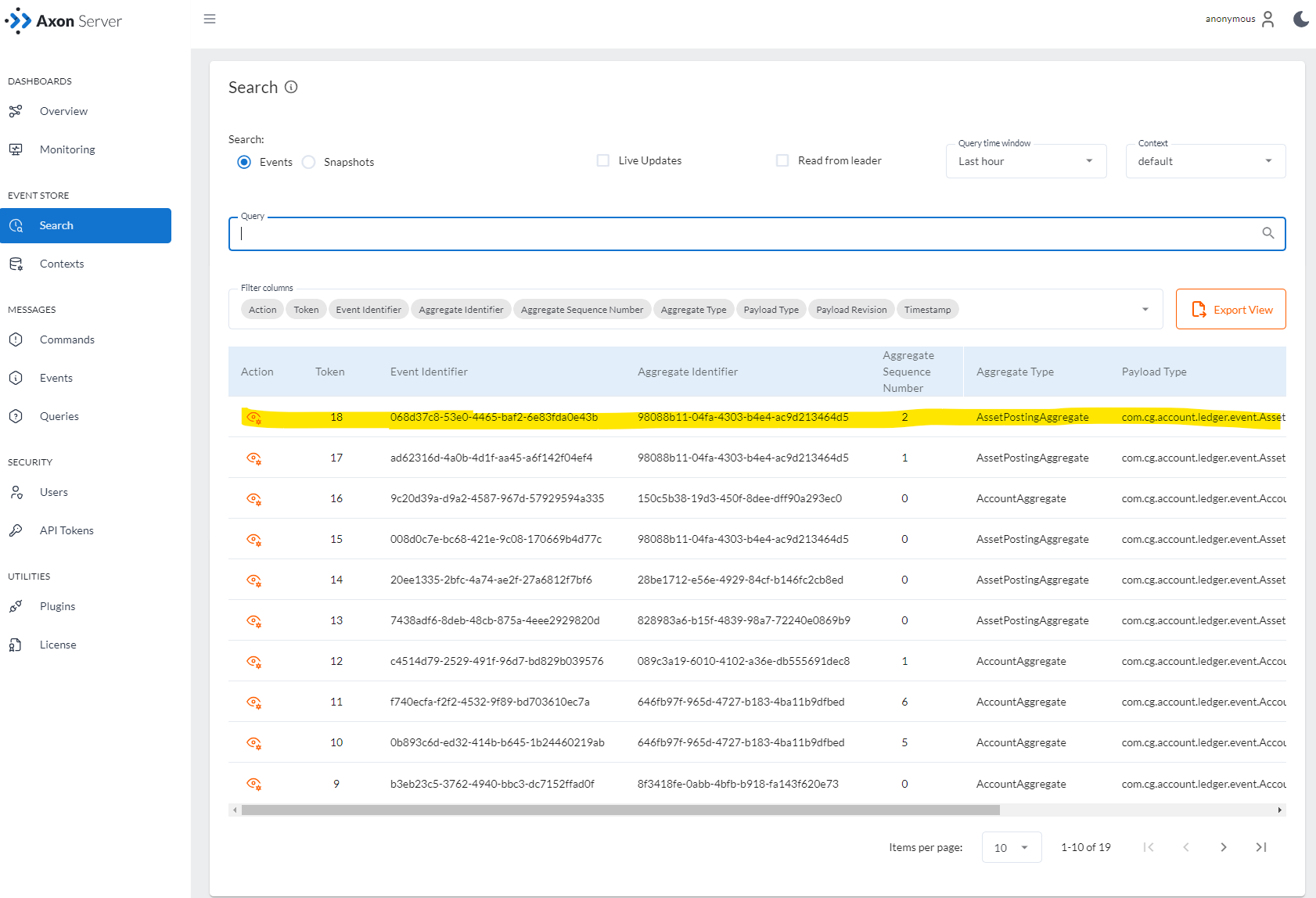
API call from postman



Status Updated

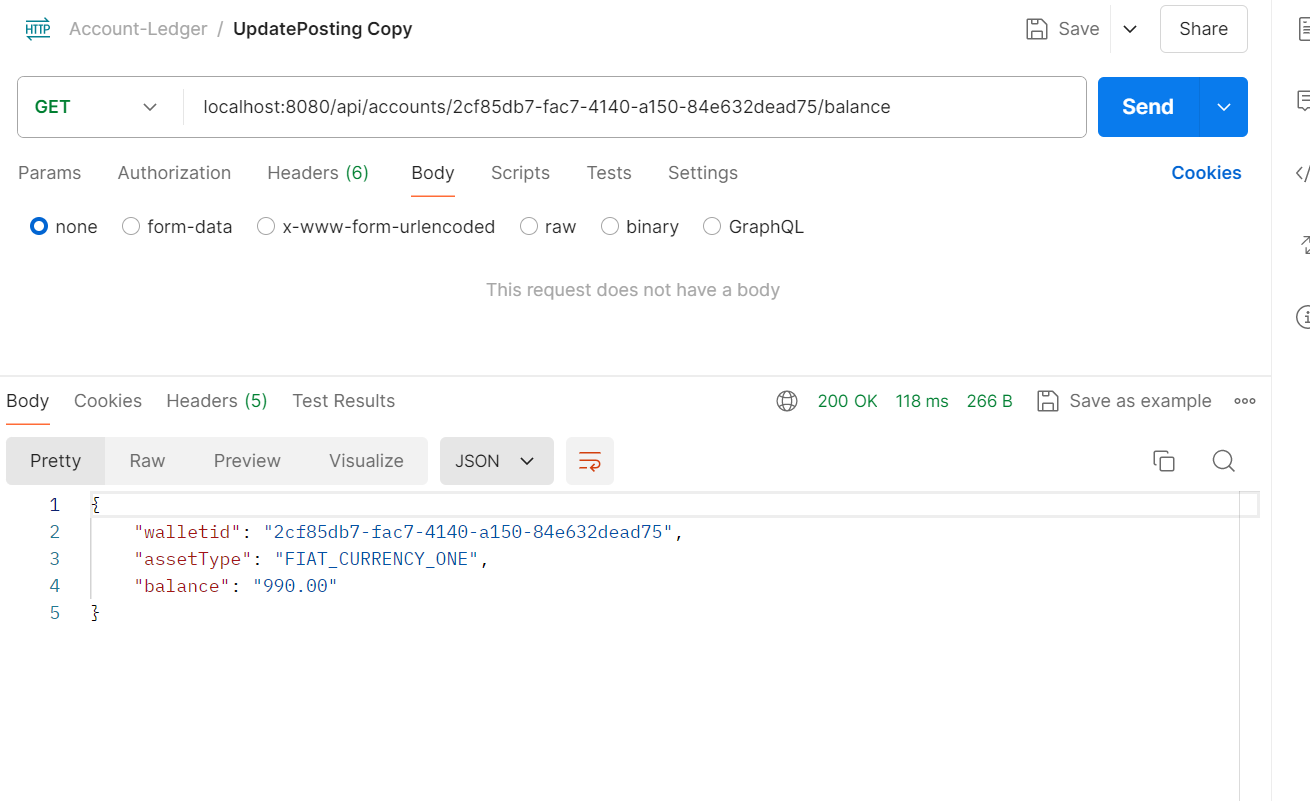


Axon Event :



### Get Wallet Balance

I have fetched balance for wallet “**2cf85db7-fac7-4140-a150-84e632dead75**”



AXON Event for Query

