📘 Project Brief: Trade Reconciliation System

🧩 Overview

The **Trade Reconciliation System** is a full-stack application designed to ingest, manage, and reconcile financial trade data from different source systems (e.g., Bloomberg, Reuters). It helps financial institutions identify inconsistencies between trade records from multiple systems to ensure data accuracy and regulatory compliance.

🏗️ Architecture

- **Backend:** Java + Spring Boot (REST APIs)
- **Frontend:** React.js
- **Database:** PostgreSQL
- **ORM:** Hibernate / JPA

🎯 Key Features

1. Trade Management

- Add, view, and delete trade records
- Store metadata like `trade_id`, `instrument`, `price`, `quantity`, `source_system`, `trade_date`

2. Instrument Management

- Manage financial instruments (e.g., AAPL, GOOG, MSFT)
- Fields include `symbol`, `name`, and `ISIN`

3. Reconciliation Engine

- Compares trades with the same `trade_id` from different systems
- Identifies mismatches in `price`, `quantity`, etc.
- Logs differences for auditing and reporting

4. Audit Trail

- All reconciliations are stored with a unique run ID
- Summary of matched/unmatched counts
- Persistent storage of differences for future lookup

🧪 Reconciliation Logic

- Group trades by `trade_id`
- If two or more versions of the same trade exist:
  - Compare each field (e.g., price, quantity)
  - If differences exist, log them with field-wise detail
- Save a `ReconciliationRun` summary entry for each session

🌐 API Endpoints

Trade APIs

- GET /api/trades
- POST /api/trades
- GET /api/trades/{id}
- DELETE /api/trades/{id}

Instrument APIs

- GET /api/instruments
- POST /api/instruments
- GET /api/instruments/{symbol}
- DELETE /api/instruments/{id}

Reconciliation APIs

- POST /api/reconciliation/start
- GET /api/reconciliation/differences/{runId}
- GET /api/reconciliation/status/{runId}

📊 UI Capabilities

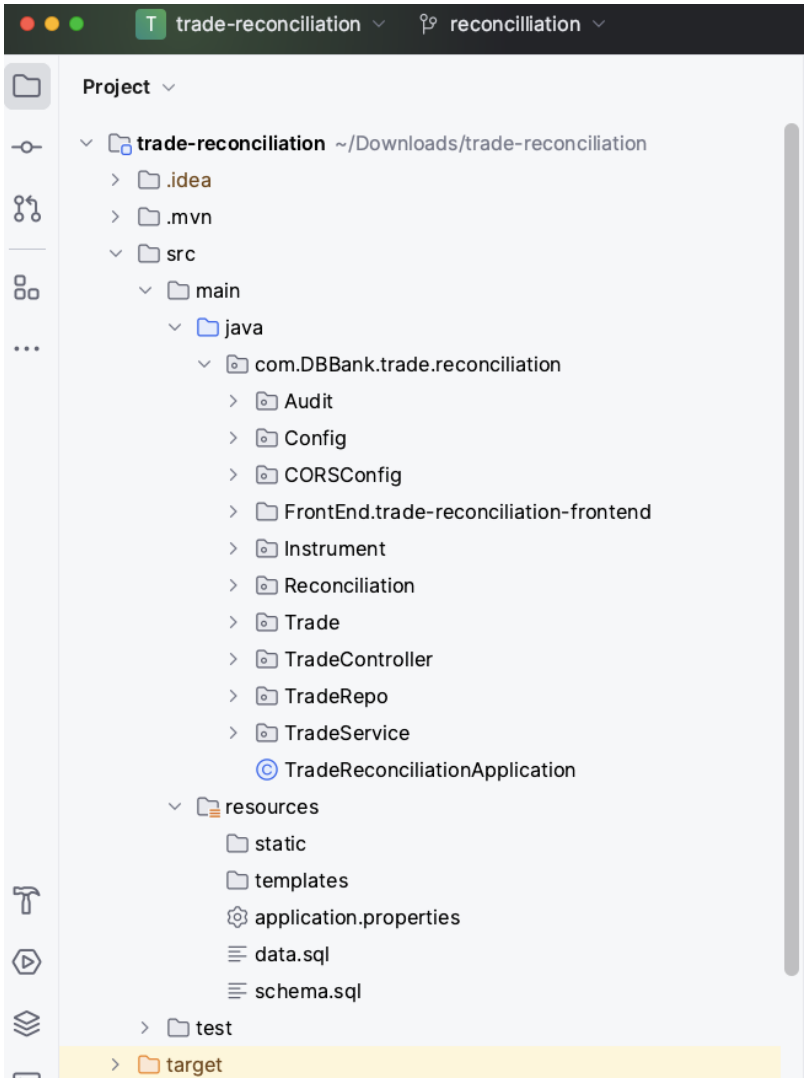
- View all trades and instruments
- One-click reconciliation trigger
- Visual summary of matched/unmatched trades
- Table showing detailed mismatches by field

📜 Use Cases

- Investment banks validating post-trade data
- Risk and compliance teams spotting data discrepancies
- Internal auditors needing reconciliation reports

🚀 Benefits

- Improved data integrity
- Easier compliance with financial regulations
- Automated and auditable reconciliation process
- Developer-friendly and extensible codebase



FrontEnd.trade-reconciliation-frontend: Reactjs Code: Port/3000

Trade-reconciliation: BackEnd SpringBoot Code: Port/9090

CORSConfig: Configuration of Cross Origin Resource Sharing: Since there are multiple Controllers, here we are creating a centralized / single file for setting up CORS

🔧 Schema Overview

1. trade

Column	Type	Description
id	SERIAL (PK)	Internal unique ID
trade_id	VARCHAR(255)	Business key, used in reconciliation
instrument	VARCHAR(255)	Symbol of instrument (AAPL, GOOGL, etc.)
price	NUMERIC	Trade price
quantity	INTEGER	Trade quantity
source_system	VARCHAR(255)	System name (e.g., Bloomberg)
trade_date	DATE	Trade date

---

2. instrument

Column	Type	Description
id	SERIAL (PK)	Internal unique ID
symbol	VARCHAR(50)	Ticker symbol (AAPL)
name	VARCHAR(100)	Full company name
isin	VARCHAR(20)	International ID

---

3. reconciliation_run

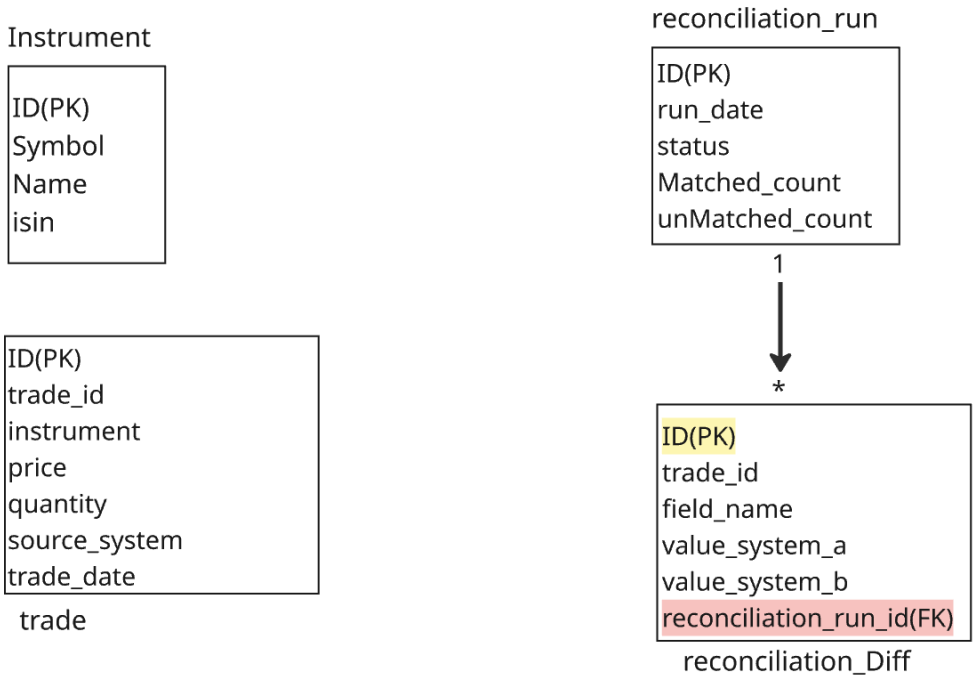
Column	Type	Description
id	SERIAL (PK)	Unique run ID
run_date	TIMESTAMP	When reconciliation was done
status	VARCHAR(20)	Status (e.g., COMPLETED)
matched_count	INTEGER	Number of matched trades
unmatched_count	INTEGER	Number of unmatched trades

---

4. reconciliation_difference

Column	Type	Description
id	SERIAL (PK)	Unique diff ID
trade_id	VARCHAR(255)	Trade ID being compared (not a FK in schema)

Column	Type	Description
field_name	VARCHAR(100)	Field that differs (price, quantity, etc.)
value_system_a	VARCHAR(255)	Value from source system A
value_system_b	VARCHAR(255)	Value from source system B
reconciliation_run_id	INTEGER	FK → reconciliation_run(id)



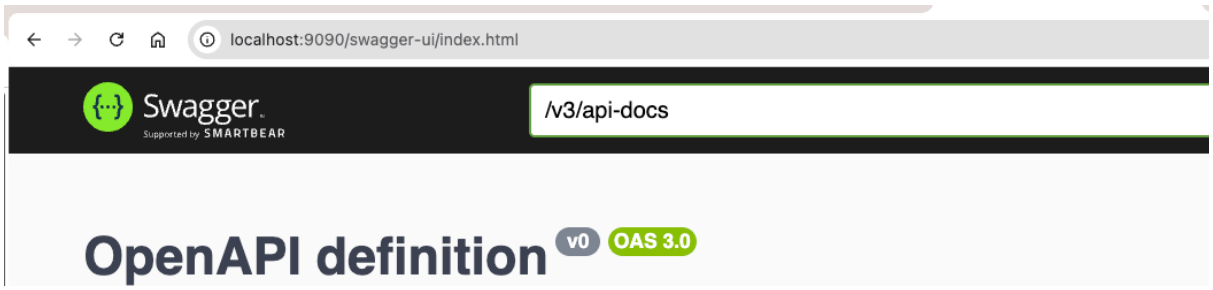
🔗 Relationships

- trade.instrument → indirectly matches instrument.symbol (not enforced by FK).
- reconciliation_difference.reconciliation_run_id → **FK** to reconciliation_run(id).

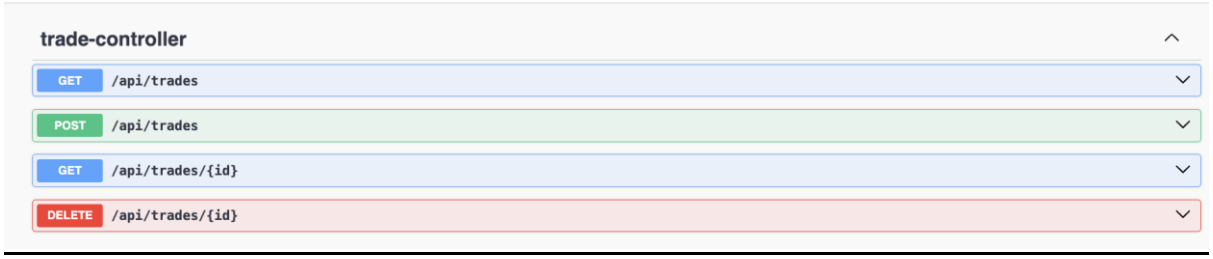
💡 Notes

- trade.trade_id is not unique because it appears from multiple systems.
- instrument.symbol is not a foreign key in trade.instrument (you could consider enforcing it for data integrity).
- No direct FK exists between trade and instrument or reconciliation_difference.trade_id and trade.trade_id, but such constraints could be added if needed.

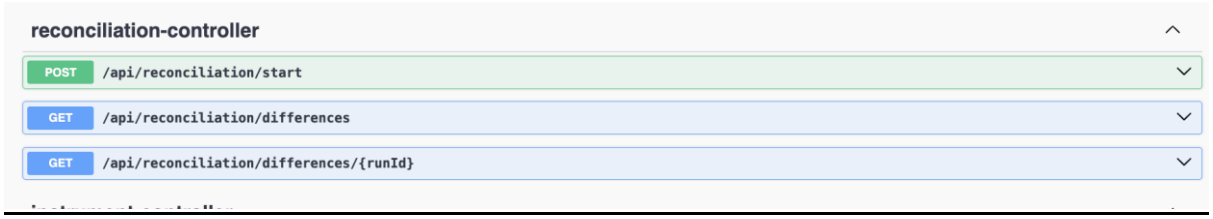
Swagger Ui:



Trade Controller:



Reconciliation-controller



Instrument-controller

Audit-controller