# Readme

## 1. Environment

Java : 17

SpringDoc: 1.7.0

Spring Boot 3.1.2

## 2. For Database

Originally, H2 is designed to be used. There is version bug with the version I used for H2. The @Qurrer doesn’t work, so I changed for oracle:

Oracle connect details are in the application.properties file.

Create table use the SQL like this:

Drop Table transaction;

**CREATE** **TABLE** transaction(

id IDENTITY **NOT** **NULL** **PRIMARY** **KEY**,

customer\_id **VARCHAR**(10),

transaction\_type **VARCHAR**(10),

transaction\_time **timestamp**,

paid\_amount **INT**,

);

Insert data from below sql script:

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1001, 'ntao', 125, 'Sale', TO\_TIMESTAMP('2023-06-22 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1002, 'ntao', 50, 'Sale', TO\_TIMESTAMP('2023-06-28 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1003, 'ntao', 180, 'Sale', TO\_TIMESTAMP('2023-07-25 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1004, 'ntao', 200, 'Sale', TO\_TIMESTAMP('2023-05-23 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1005, 'ntao', 500, 'Sale', TO\_TIMESTAMP('2023-06-11 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1006, 'shuying', 150, 'Sale', TO\_TIMESTAMP('2023-06-24 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1007, 'shuying', 250, 'Sale', TO\_TIMESTAMP('2023-05-20 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1008, 'shuying', 400, 'Sale', TO\_TIMESTAMP('2023-07-12 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1009, 'shuying', 200, 'Sale', TO\_TIMESTAMP('2023-07-22 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1010, 'lee', 130, 'Sale', TO\_TIMESTAMP('2023-05-22 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1011, 'lee', 120, 'Sale', TO\_TIMESTAMP('2023-07-25 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1012, 'lee', 160, 'Sale', TO\_TIMESTAMP('2023-06-10 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

**INSERT** **INTO** transaction (id, customer\_id, paid\_amount, transaction\_type, transaction\_time) **VALUES** (1013, 'lee', 300, 'Sale', TO\_TIMESTAMP('2023-06-22 17:16:05', 'YYYY-MM-DD HH24:MI:SS'));

## 3. Logging implementation

Use slf4j and log4j together. Only implemented in the PromotionReportService class.

## 4. Http response status

Http response status are implemented with SpringDoc dependency.

## 

## 5. Unit test is planned to use Junit 5 and Mokito

Due to the tight time line, no meaningful test is done. Only implemented in two locations.

## 6. API E2E Browser Test

With end point /promote/points-report at localhost:8080

We can get following:

