# **Rewards Program Application Documentation**

# **Table of Contents**

## Project Overview

## Technologies Used

## Setup Instructions

### Prerequisites

### Clone the Repository

### Build the Project

## Configuration

### application.properties

### data.sql

## Running the Application

## Usage

### H2 Console

## Endpoints

## Testing

## License

# **Project Overview**

The Rewards Program Application is a simple Spring Boot application that manages customer rewards and transactions. It uses an H2 in-memory database for quick data storage and retrieval, making it ideal for development and testing purposes.

**Technologies Used**

**Java 21: Programming language used for development.**

**Spring Boot 3.1.2: Framework for building the application.**

**Spring Data JPA: For interacting with the database using JPA.**

**H2 Database: In-memory database for development.**

**Maven: Build tool for managing dependencies.**

## **Setup Instructions**

**Prerequisites**

* Java Development Kit (JDK) 21 or above installed on your machine.
* Maven installed for dependency management and building the project.
* An IDE like IntelliJ IDEA or Eclipse for development (optional).

**Clone the Repository**

To get started with the project, clone the repository from GitHub:

git clone https://github.com/yourusername/rewards.git

cd rewards

**Build the Project**

Navigate to the project directory and run the following command to build the project:

mvn clean install

## **Configuration**

## **application.properties**

The application.properties file contains the configuration for the application, including database settings and JPA properties. Here’s a sample configuration:

# Application name

spring.application.name=rewards

# H2 Database configuration

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

# JPA and Hibernate configuration

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=create-drop

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

# H2 Console configuration

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

# SQL initialization configuration

spring.datasource.initialization-mode=always

spring.sql.init.mode=always

spring.sql.init.platform=h2

# Logging configuration for SQL

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

**data.sql**

The data.sql file is used to initialize the database with tables and sample data:

-- Create the CUSTOMER table

CREATE TABLE CUSTOMER (

id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

name VARCHAR(255) NOT NULL

);

-- Create the TRANSACTION table

CREATE TABLE TRANSACTION (

id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

date DATE NOT NULL,

amount DOUBLE NOT NULL,

customer\_id BIGINT,

FOREIGN KEY (customer\_id) REFERENCES CUSTOMER(id)

);

-- Insert sample data into CUSTOMER table

INSERT INTO CUSTOMER (name) VALUES ('John Doe');

-- Insert sample data into TRANSACTION table

INSERT INTO TRANSACTION (date, amount, customer\_id) VALUES ('2023-01-15', 120, 1);

INSERT INTO TRANSACTION (date, amount, customer\_id) VALUES ('2023-02-20', 75, 1);

INSERT INTO TRANSACTION (date, amount, customer\_id) VALUES ('2023-03-10', 200, 1);

# **Running the Application**

To run the application, use the following command:

mvn spring-boot:run

Once the application is running, you can access the H2 console at:

**URL**: http://localhost:8080/h2-console

**JDBC URL**: jdbc:h2:mem:testdb

**Username**: sa

**Password**:

## **Usage**

### **H2 Console**

The H2 console allows you to interact with the in-memory database. Use the console to execute SQL queries and view data in your tables.

## **Endpoints**

The current implementation does not expose any REST endpoints. Future iterations may include endpoints for managing customers and transactions.

## **Testing**

To run unit tests, use the following command:

mvn test

## **License**

This project is licensed under the MIT License. See the LICENSE file for details.