Trade Store App Documentation

Summary:

This document describes in detail of Trade store app. With this app client will be able to store Trade data or retrieve data with the help of available API. Trade app also takes care of updating expired trades with scheduler on every day at 1.00 AM.

This app is implemented with Java 8, Spring Boot, Maven, H2Database.

API's Implemented:

- GET /trades
- GET /trade/{id}
- PUT /trade{id}
- GET /ping

API Endpoints:

Http Method	URL	Description
Get	http://localhost:8080/trade/T1	Get Trade by ID
Get	http://localhost:8080/trades	To Get all Trades
Put	http://localhost:8080/trade/T1	To Update to create Trade
Get	http://localhost:8080/ping	To check the health of the app

Authentication Used:

Authentication used in trade store app is Basic Authentication. API's can be accessed with below users.

Username	password
admin	admin
user	user
swagger	swagger

Important URL:

Swagger Url: http://localhost:8080/swagger-ui.html#

H2DB Console url: http://localhost:8080/h2-console/
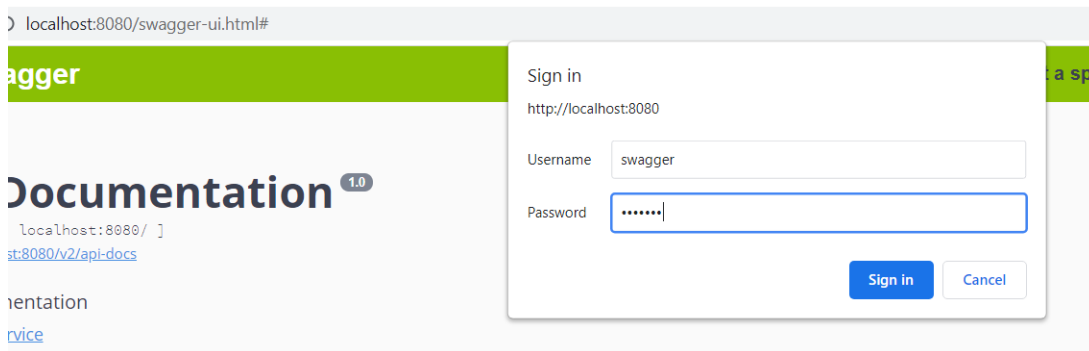
How to connect to Swagger:
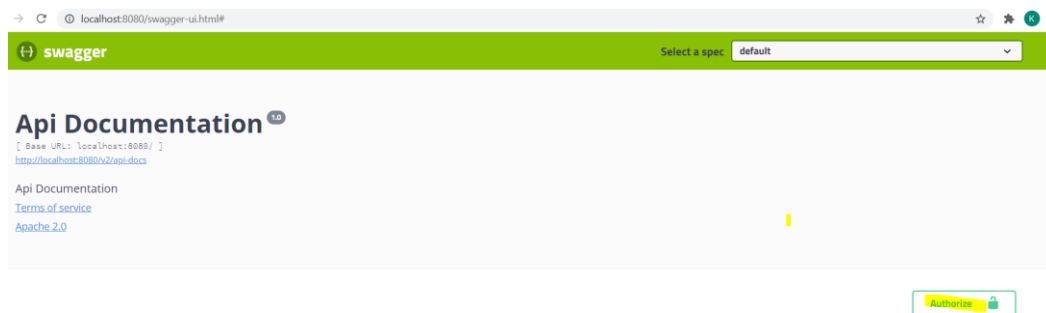
For opening Swagger user need to provide username and password as shown in below screen.

To use swagger to call API, user can authorize in the swagger.

1. Click on Authorize highlighted in yellow.

2. provide user name and password and then click on Authorize.

3. when the request is authorized.

How to Connect to H2DB:

On click on H2DB console in browser, cline needs to pass below details.

Driver: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

Username: sa

Password: ""(empty)

Reference:

Trade Store App Documentation

Tables created:

Below Tables are created for Trade store app

Table Name	Remarks
Trade	Holds trade data
Users	Holds user data
Roles	Holds user role data

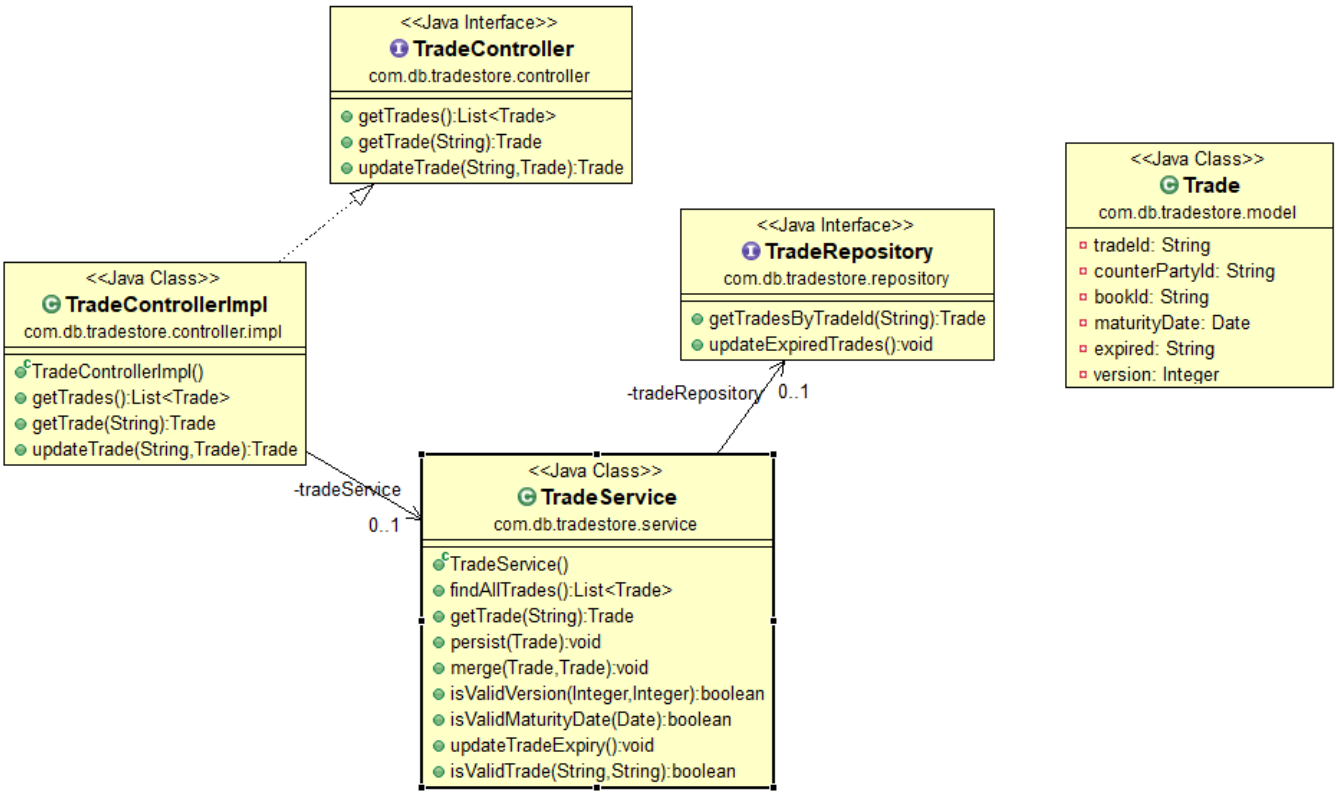**Test Cases:**

Below Junit test cases has been written which covers most of the scenarios.

✓ TradeStoreAppApplicationTests 1 sec 14 ms
   ✓ getAllTrades() 551 ms
   ✓ getTradeById() 201 ms
   ✓ updateTradeOnIdMismatch() 11 ms
   ✓ updateTradeOnInvalidVersion() 21 ms
   ✓ updateTradeInvalidMaturityDate() 17 ms
   ✓ updateTrade() 197 ms
   ✓ contextLoads() 16 ms

UML Diagram:

Trade Store App Documentation

<<Java Interface>>
① TradeController
com.db.tradestore.controller

- getTrades():List<Trade>
- getTrade(String):Trade
- updateTrade(String,Trade):Trade

<<Java Class>>
④ TradeControllerImpl
com.db.tradestore.controller.impl

- TradeControllerImpl()
- getTrades():List<Trade>
- getTrade(String):Trade
- updateTrade(String,Trade):Trade

<<Java Interface>>
① TradeRepository
com.db.tradestore.repository

- getTradesByTradeId(String):Trade
- updateExpiredTrades():void

<<Java Class>>
④ Trade
com.db.tradestore.model

- tradeId: String
- counterPartyId: String
- bookId: String
- maturityDate: Date
- expired: String
- version: Integer

-tradeRepository 0..1

-tradeService
0..1

<<Java Class>>
④ TradeService
com.db.tradestore.service

- TradeService()
- findAllTrades():List<Trade>
- getTrade(String):Trade
- persist(Trade):void
- merge(Trade,Trade):void
- isValidVersion(Integer,Integer):boolean
- isValidMaturityDate(Date):boolean
- updateTradeExpiry():void
- isValidTrade(String,String):boolean

Thank You 😊