Stock Exchange Simulator Documentation

Contents

[What is JPA? 1](#_Toc80731669)

[What is Hibernate? 2](#_Toc80731670)

[What are JSON Web Tokens? 2](#_Toc80731671)

[Spring Initializr 4](#_Toc80731672)

[Set up: 4](#_Toc80731673)

[https://start.spring.io/ 4](#_Toc80731674)

[Spring Boot 4](#_Toc80731675)

[Application. Properties 4](#_Toc80731676)

[PostgreSQL 4](#_Toc80731677)

[React 5](#_Toc80731678)

[Set up: 5](#_Toc80731679)

# What is JPA?

JPA is Java Persistence API, which is a specification provided to access, manage, and persist data between Java object and relational databases in Java applications. Spring Data JPA is implementation to implement JPA based repository and remove lots of boiler code. It focuses on using JPA to store relational data in a relational database while giving us access and persist data between Java object/class and relational database.

## @Entity

Indicates that it is a JPA entity, which is a persistence object stored as a record in the database.

## @Table

Indicates that this JPA entity is mapped to a table named…

## @Id

Indication so that JPA recognizes it as the objects ID

## @GeneratedValue

Annotated with @Id to indicate the ID should be generated automatically.

## @Query

Is an interface that is implemented by each JPA vendor to obtain related objects that meet the criteria

# What is Hibernate?

Hibernate is a lightweight, open-source ORM tool that is used to store Java objects in the relational database system. It is a provider of JPA. It provides implementation of classes and uses its own query language called HQL (Hibernate Query Language).

# What are JSON Web Tokens?

JWT is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. JWT consists of three parts separated by dots ( . ), which are: Header, Payload, Signature.

## Header:

{

“algorithm”: “HS256”,

“type:”: “JWT”

}

## Example Payload:

{

“name”: “Nick Rudolph”

“admin” true

“subject: (user ID)

“expriesAt”

“issuer”: (the entity to generate and issue the JSON Web token)

}

## Signature

{

Sign -HMACSHA256 (base64 URL encoded header, base64 URL encoded payload, secret)

}

## Tokens:

Upon initial authentication, the user will receive two tokens.

## Access Token:

Your typical JSON Web Token that is sent with every request.

## Refresh Token:

This special kind of token is persisted in a database, mostly owned by an Authentication Service. Once the access token expires, the front-end will send a refresh request with the refresh token. The authentication server will generate a new JWT Access Token and send it back to the user.

File Structure

Config

Controller

API Layer 🡪 Service Layer

@RestController

Defines the RESTful web services. It servers JSON, XML, and custom response.

@RequestMapping

Defines the Request URL to access the REST Endpoints.

@GetMapping

@PostMapping

Model

Class

Class Attributes

Class constructors

Setters & Getters

@Entity

Specifies that the class is an entity and is mapped to a database table

@Table

Specifies the table name of the database

@Id

Specifies the primary key of an entity

@SequenceGenerator

@Generated Value

Give the generation strategy for the values of the primary key

Repository

Data Access Layer

Encapsulates storage, retrieval, and search behavior

Extends from the JpaRepository

@Repository

Responsible for data access. Performs any operation on the object.

@Query

Service

Service Layer 🡪 Data Access Layer

@Autowired

Autowire the object of the repository inside the service class to invoke their methods

Security

# Spring Initializr

## Set up:

## <https://start.spring.io/>

A screenshot of a computer

Description automatically generated with medium confidence

# Spring Boot

## Application. Properties

spring.datasource.url=jdbc:postgresql://localhost:5432/users  
spring.datasource.username=postgres  
spring.datasource.password=Root38!  
spring.jpa.hibernate.ddl-auto=create-drop  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
spring.jpa.properties.hibernate.format\_sql=true

# PostgreSQL

CREATE DATABASE users;

CREATE TABLE person (

userID BIGSERIAL NOT NULL PRIMARY KEY,

firstName VARCHAR(100) NOT NULL,

lastName VARCHAR(100) NOT NULL,

email VARCHAR(254) NOT NULL,

dateJoined date NOT NULL

);

# React

## Set up:

The entire React app is under the folder Stock/src/main/ui using command:

npx create-react-app ui

npm install axios

npm i @chakra-ui/react @emotion/react@^11 @emotion/styled@^11 framer-motion@^4

npm install react-router-dom

npm install @chakra-ui/react

npm install @chakra-ui/gatsby-plugin

npm install react-bootstrap@next bootstrap@5.0.2