Spring Boot Learning

Discussion Topics:

- Spring Boot and Its Annotations
- Embedded Database with Spring Boot
- Spring with JavaFx

Spring Boot

Spring Boot is an open source Java-based framework. Spring Boot provides a good platform to develop a stand-alone and production-grade application.

We choose Spring Boot because of the features and benefits it offers as given here −

- In Spring Boot, most things are auto configured.
- It provides and manages REST endpoints.
- It offers annotation-based coding.
- Eases dependency management.

Ref: https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm

Spring Annotations

- @SpringBootApplication
- @Controller
- @Service
- @AutoWired
- @Configuration
- @EnableAutoConfiguration
- @RestController
- @RequestMapping
- @MessageMapping
- @Scheduled
- @Repository
- @Entity
- @Column
- @Table

Ref: https://www.javatpoint.com/spring-boot-annotations

Embedded Database

Embedded database as in databases that do not need a server, and are embedded in an application. This means everything is managed by the application. Some embedded DBMS are –

- Apache Derby
- H2
- HSQLDB
- Firebird

Apache Derby

Derby is an embeddable SQL engine written entirely in Java. Fully transactional, multi-user with a decent SQL subset, Derby is a mature engine and freely available under the Apache license and is actively maintained.

Hibernate

Hibernate is an object-relational-mapping (ORM) tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. It also create-update-drop database table according to domain models.

Hibernate's primary feature is mapping from Java classes to database tables, and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from the manual handling and object conversion of the result set.

Hikaricp

HikariCP is a reliable, high-performance JDBC connection pool.

Ref: https://www.baeldung.com/hikaricp
https://www.baeldung.com/spring-boot-hikari

Named Query

Sql query using declarative pattern. Such as –

- `findByName`
- findAllOrderByDateDesc

Ref: https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html

schema.sql

‘schema.sql’ is a sql script file. Spring boot can create database schema a defined in schema.sql file. In auto-configuration mode hibernate handle this work.

data.sql

‘data.sql’ is a sql script used to load setup/configuration tables on application startup.

Spring with JavaFx

- ➢ Create a Spring Boot Application
- ➢ Add JavaFx dependency in the pom.xml
- ➢ Add bellow arguments to VM Options

```
--module-path  /home/shazid/javafx-sdk-14.0.1/lib
--add-modules javafx.controls,javafx.graphics,javafx.fxml
--add-exports  javafx.graphics/com.sun.javafx.sg.prism=ALL-UNNAMED
```

- Spring Boot Application **Main** method extends Javafx **Application**
  ```
  public class Main extends Application() { }
  ```
- Have to create a JavaFx module.info and add required module/dependency in the module.info.
- Create a Navigation class which will manage our view scenes and instantiate right controller using Spring Dependency Injection.
- We will use Spring Boot DI for Service injection.
- Here we use embedded Apache Derby database as before.

Problems and Learnings

1. Manually configure derby with spring boot
   - Should do minimal configuration.

```
spring.datasource.jdbc-url=jdbc:derby:testdb;create=true
spring.datasource.username=derbydb
spring.datasource.password=derbydb
spring.datasource.driver-class=org.apache.derby.jdbc.EmbeddedDriver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.DerbyTenSevenDialect
spring.jpa.hibernate.ddl-auto=update
```